IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA
RICHMOND DIVISION

TRUSTEES OF COLUMBIA UNIVERSITY IN THE
CITY OF NEW YORK

Plaintiff;

v.

Civil Action
3:13CV808

SYMANTEC CORPORATION,

Defendant.

September 4, 2014
Richmond, Virginia
9:00 a.m.

MARKMAN HEARING

BEFORE: HONORABLE JAMES R. SPENCER
United States District Judge

APPEARANCES: DANA D. McDANIEL, ESQ.
GAVIN SNYDER, ESQ.
JASON G. SHEASBY, ESQ.
RICHARD M. BIRNHOLZ, ESQ.
DAVID I. GINDLER, ESQ.
Counsel for Plaintiff;

DABNEY J. CARR, IV, ESQ.
DAVID A. NELSON, ESQ.
ALEXANDER RUDIS, ESQ.
NATHANIEL A. HAMSTRA, ESQ.
Counsel for Defendant.

JEFFREY B. KULL
OFFICIAL COURT REPORTER

P-R-O-C-E-E-D-I-N-G-S

THE CLERK: Case Number 3:13CV808: Trustees of Columbia University in the City of New York versus Symantec Corporation. The plaintiff is represented by Dana McDaniel, Gavin Snyder, Jason Sheasby, Richard Birnholz, and David Gindler. The defendant is represented by Dabney Carr, David Nelson, Alexander Rudis and Nathaniel Hamstra. Are counsel ready to proceed?

MR. NELSON: We are ready, Your Honor.

MR. McDANIEL: We are ready. Your Honor, I'll just take a brief second. Dana McDaniel on behalf of Columbia University. And I wanted to introduce the folks who will be presenting today: Jason Sheasby, who has been previously introduced to the Court, Gavin Snyder, and Rich Birnholz. With them at counsel table are David Gindler and Xinlin Lee. Here on behalf of Columbia University is Jeffrey Sears, their Associate General Counsel and Chief Patent Counsel. Thank you, Your Honor.

THE COURT: All right. Mr. Carr?

MR. CARR: Good morning, Judge. On behalf of Symantec today, speaking will be David Nelson, sitting here, Alex Rudis, and Nathan Hamstra, Nate Hamstra. And with us in the courtroom we have from Symantec David Majors. Thank you, Your Honor.

THE COURT: All right. Let's get started.

Trustees of Columbia?

MR. SHEASBY: Your Honor, the structure the parties have agreed to today is, both will give a short introduction; after that we will go patent family by patent family; and within the patent family we will go term by term ping-ponging. We will present our construction of a term, Columbia, and Symantec will respond to that, and we will bring it to a head at that point if Your Honor is agreeable with that.

THE COURT: That's fine.

MR. SHEASBY: Madam Courtroom Officer, could I have the slide presentation?

So Your Honor, there are three families of patents that are at issue in the litigation. The first family of patents relates to the analysis of executable e-mail attachments. We will hear a lot about that today, but in terms of claim construction, the central issue that we will be discussing is the type of information that is extracted from an executable. An executable being the attachment that you receive on an e-mail. You extract very particularized types of information from that e-mail, and by doing that you are able to make determinations as to the danger associated with that e-mail. What you extract is described as a feature. We will hear a lot about features discussed today.

The second family of patents relates to detecting anomalous registry accesses. That family of patents, the '084 family of patents involves an analysis of a model and divergence from that model to detect something that's potentially malicious.

The third family of patents, the '115 family of patents, relates to anomaly detection as well, detection of something that diverges from normal. But that patent focuses on, for purposes of claim construction, a very unique process for selectively analyzing portions of the code for efficiency purposes, and then a type of distributing computing that allows for the efficient updating or creation of models.

So all three families come out of the work of Professor Stolfo and Professor Keromytis. They are Columbia professors. They have been Columbia professors for many, many years. Professor Keromytis is now on leave, now at the National Science Foundation and soon to move to the Department of Defense where he is doing computer security research.

What I want to start with, I want to set the stage in terms of the technology. What I mean by that is I want to talk about the state of the art in which the inventors were working. So historically, the standard way of detecting malicious programs, detecting danger, was known as a signature list. The focus was exclusively on what was evil, learning and cataloging all the evil that had appeared before, and making sure that you kept an exhaustive list of that evil so that when you saw it again, you could stop it from ever hurting you. In other words, a signature list was created.

And the signature list is very much like a thumbprint, a fingerprint, like an officer would use to log and catalog a criminal. Of course, your logging and cataloging of criminals was only as good as your list of fingerprints. Here is a classic example of how signature matching would work. So you have a signature of a malicious program, and you are looking for programs that are coming across your computer. And when one matches the signature, you know there is a danger.

Now, the failure of this type of system, because it is so focused only on that which was evil, that which was bad, is it had no way of understanding and dealing with new evils that were to appear, new malicious programs that were different than what had come before. They were no less malicious, but they didn't have the same fingerprint, they didn't have the same thumbprint. These are sometimes referred to as Zero-Day attacks. The ultimate failure of this focus exclusively on that which was, focus exclusively on that which was evil was that you would never be able to use the signatures to detect that which was new.

So this is an example of the failure. So all these are new viruses. And they are slightly different from what had come before. And, of course, they can't be detected because the signatures don't match.

Well, a number of research laboratories across the country, led by the Columbia research laboratories, began experimenting with the use of machine learning to try to deal with these Zero-Day attacks, to depart from this exclusive focus on only that which was bad. Machine learning really has three pieces to it. The first is you collect a massive amount of data. You then train a model with the data. Training a model, creating a model, in this case, for example, a model of that which is normal so that you can detect divergences from it, for example, you apply that model.

So I'm not going to talk about it in the computer science context to begin with. I'm going to talk about it in an abstracted way. Imagine you want to create a model of apples. I want a model of what it is to be an apple. Because when I encounter fruit, I want my machine to pick out when I encounter something that's not an apple because all I want is apples. I only want apples and I want to exclude everything else that's not an apple. What you can do is you can extract features from that apple. What are the features of the apple? Well, its shape. It is round. What are the features of the apple? It has seeds. What are the features of the apple? It grows on a tree. These features that you extracted from the apple allow a computer to begin learning what it means to be an apple and what it means to not be an apple.

Now, there is a problem with constructing your model of apples if you focus exclusively and solely on apples. That is, the features you select to define that apple may be features that are shared by other fruits. For example, oranges. They have a rounded shape, they have seeds, they grow on a tree. And so focusing slowly on those three features from the apple, what would actually create a bad model, a model in which you treat an orange as an apple, because the features match. But what you can do is by extracting as much information as you can about what makes an apple an and he will and how an apple is different from an orange, for example, you can yes create a more robust model. Don't just focus on the shape, don't just focus on the seeds, don't just focus on what grows on the tree. Focus on the color. Oranges's color is different from apples. Oranges are red. Focus on the thickness of the skin, the juice content. All of these are features that tell you about what makes something an apple and what makes an apple different from other fruits. The idea is not to blind yourself to data.

So after collecting all this data, you can begin to train a model. You can begin to construct a model that allows your machine to start to think, to start to distinguish from its natural environment. So when the machine encounters a fruit that it has never seen before, it will say what's its shape? Is it round or is it crescent? Does it have seeds? Does it grow on a tree? What color is it? What does its skin look like, thick or thin? What's its juice content? These features that have been extracted from the data allow the machine to create a model that defines what it is to be an apple and defines what it means to be not an apple or to be different from an apple.

And so if you do this analysis, you can detect an apple. And what's elegant about this is this analysis can actually exclude things that maybe look like apples in certain ways, have some of the same features, but are different. So for example, a litchi fruit. A litchi fruit has many of the characteristics of an apple that I list up here. It's round, it has seeds, it grows on trees, it has red skin, but it is very thick skin, not a thin skin. And so this process of extracting as many features as you can to learn not just what makes an apple an apple but how apples are different from other fruits by actually considering other fruits in the training process, you can create a model of appleness.

Now, I talked about apples and oranges. But I want to now bring it back to what we are focused on, which is computer learning. So instead of extracting characteristics of fruits, think of extracting thousands and thousands of features from normal malicious programs. You create this massive dataset of how normal programs act, how malicious programs act, and how that distinction creates meaning.

You use these features to train a model, create a model that detects divergence from normal. You obviously have to start with the apple. You have to start with that which is normal, because by doing so, you begin to create your model. But you don't blind yourself to everything that exists. The whole essence of the machine learning process that Professors Stolfo and Keromytis described is this massive extraction of information. So the three families of patents that we are going to be speaking about all relate to this machine learning strategy at a very basic level. But they all present very, very important improvements over machine learning to make as efficient, as accurate, as fast models as possible to detect that which is dangerous. And that's what we are going to be discussing today, Your Honor.

So at this point, I believe Symantec is going to give an opening presentation as well, and then we will turn to the individual patents. Is that correct, Mr. Nelson?

MR. NELSON: Yes.

THE COURT: All right.

MR. NELSON: Thank you, Your Honor. David Nelson on behalf of Symantec. Good to see you again.

THE COURT: Good to see you.

MR. NELSON: So we don't have a big problem with what we just heard. I don't think that's really where the dispute lies. There is one additional fact, I think, that counsel contrasted what the claimed inventions are in the three families of patents here to signatures, looking for the actual fingerprints and things you have seen before. And although the patents do discuss that, it is not quite accurate to lay the groundwork to say that what we are looking at now is an invention that in the past only these signature systems exist.

There is another very important type of system that existed, and they call these misuse systems. In fact, they are called in the patents, there's various citations we have in the brief talking about that. And these misuse systems, what they did, for example, was they collected information on bad activities, the kinds of things that bad programs might do. For example, like if you have a program that comes on to a machine and immediately copies all of the contents of another file into its program space that it is operating on, things like that that normal programs really don't do. And those kinds of systems existed out there, no question about it. Frankly, the kinds of systems we are talking about looking for normal behavior existed as well, but I'm just talking about now for purposes of MARKMAN what the patents describe and the environment we are looking in..

So those kinds of systems that knew about bad behavior and looked for patterns of bad behavior, you know, for example, if you liken this to a home alarm system, you know, somebody that's going to protect security in your house, say, "Well, wait a minute, if somebody, if I see somebody climbing over the fence at night or somebody trying to open a garage at 2 o'clock in the the morning," or something like that, "these are things that we have looked at and decided those are bad behaviors. We will monitor for those things and if we see them we will assume that whatever is trying to do those is acting maliciously."

That's not what these patents are about. And the patents themselves say that. It is an important thing when we get into some of the claim construction issues later in the case. So I just want to lay that additional piece of foundation, but not really dispute a lot of what you heard from counsel. I don't think that's where our dispute really lies. Rather, and let me go to the particular patents. If I could have Slide 2 of our presentation. And we have copies of these we will pass up to Your Honor here before we get into it, really start going through the claim construction issues. But that has to do with, the claim construction issues are really what systems did you claim, right? What we are seeing from Columbia, we believe, and you see some of this reflected in the brief, is an attempt to say, "Well, we have claimed the whole universe. We have claimed the general category of these things." And when we get into these claim construction issues you will see that's not the case, that we are talking about much more specific things, which is typically what you see out of patents.

So in this first family of patents that counsel described, that's the '544, ''907, that's the one about filtering e-mail attachments. We will get into those. I think it is actually the first set. There, at least in my mind from going through the brief, the real issue there is counsel said you are going to go in, look at the attachments, there are executable, things that the computer can run, that are attached to e-mails. And you are going to go in and look at particular features, like for example with a person, you could look at their features, say, well, for the person we are going to look at their hair color, their eye color and how tall they are, various things like that. And we can just have those be features of a person. So you can do the same thing, according to these patents, with e-mail attachments. I'm going to go in and look at various components of these executables and pull out information so that I can create this basically list, this category of features.

The real claim construction issue there is the claims themselves, at least the claims that are asserted here, they focus on a particular type of feature. So sticking with my example, you know, hair color would be a particular type of feature if I'm looking at a person. That's what these claims really focus on. But what we are seeing from Columbia is an attempt to rewrite that claim and say, "No, no, no, what we claim there is not a particular type of feature; it is all features in general." And that's the real issue that we are going to see with respect to the claim construction issue in that patent and that's what I, at least from our side, that's what we are going to focus on primarily.

Now, the next set of patents, if I go to Slide

21, this is the '084 and '306. Counsel described these, these are the ones where you create this model of normal behavior, you know, I am going to observe what a computer system does for, you know, take a bunch of data, a lot of data, and I'm going to create a model. This is normal. This is what it does. So when I start monitoring I'm going to look and see, wait a minute, what it is trying to do now, what something is trying, in this particular one, access the operating system registry, for example, that's not normal. That's something strange. So I'm going to assume that that's bad behavior and I'm going to shut that down.

So the main dispute that we have there is, what data do I use to create this model? Right? Is it a model of normal behavior, like the patents say, and we will get into these details, in other words, it doesn't include attacks. For example, if the system is out there running when I'm trying to collect this data, it is being attacked a bunch of times, then that data would not be data about the normal operation system because it is being attacked. So that really becomes the claim construction dispute, is, well, this model, the primary one, there's a couple other issues that are related, but the primary one, does this model that we are talking about, in order to create the normal baseline, is that normal data, attack-free data, or does it include attack data as well? That's going to be a primary issue with respect to the claim construction on those two patents, Your Honor.

Now, the last family of patents, this is Slide 53, these are the '115 and '322, I'm not sure why I give the numbers because I've been living with this case for eight months, nine months now, I can't remember the numbers. I remember them by what these patents do. So this one, as counsel described, is the one where I'm looking at a particular program and what the calls that are made to that particular program, function calls that are made to or by that particular program, you know, things that it is trying to do. And what is the normal activity again there, what does this program normally try to do? What calls are normally made and what calls are normally made to it? And here, a primary focus, I mean, one is the issue I just talked about with respect to the other patent is what data goes into creating the baseline that we decide what is normal.

But there's another issue, too, because when I get these programs, let's say I get an unknown program, Your Honor, that I don't know something about. Well, if I just let it run on the system, you know, have access to your computer basically, and it is malicious, something bad could happen to your system. It could wipe out your hard drive, take all your passwords for your bank accounts, whatever, that kind of thing, before I catch it and know that it is bad. So one of the things that I might want to do is run this on an emulator. In other words, so the program thinks it is running on the system, that it has access to all the system resources, the computer, but it really doesn't, because I've created this virtual environment, fake environment, that kind of thing. So the program thinks it's running on the system when it really isn't. An example of this might be back in the old days when WordPerfect, for example, was a word processing program, only ran on certain types of systems, wouldn't run on the Apple McIntosh platform, and so you would create an emulator so it would emulate a Windows system, and so now you could run that program because that program thought that it was running on a Windows system instead of the Apple McIntosh system. So that, it becomes an issue in this patent as well, is what is an emulator. And that's going to be the real definition. What is an emulator.

We believe Columbia is trying to define what you might use an emulator for, but not what an emulator is. And so that is going to be, in addition to this idea of what data is used to create the model, that's going to be an issue for claim construction with respect to these two patents, Your Honor.

So with that, we will get right into it and get into the terms and start getting you the information, Your Honor.

THE COURT: All right. Thank you very much.

MR. BIRNHOLZ: Good morning, Your Honor. Richard Birnholz of Irell & Manella.

THE COURT: Good morning.

MR. BIRNHOLZ: I have a set of the slides for the '544 and ''907 patent, also a copy of the introduction slides. If I could hand them to the security officer.

THE COURT: Sure.

MR. BIRNHOLZ: I've given a copy to opposing counsel.

So Your Honor, of course, is free to follow along on the book or the screen or both, whatever your pleasure. I'm going to be talking this morning about the '544 and '907 patents. Now, this family of patents, while in the same general space as the other patents, has some different characteristics. Now, this patent is called System and Methods for Detection of New Malicious Executables. So I use the '544 patent as the base reference, because the '905 patent is a continuation, so the disclosure is the same, although the claims are slightly different. But for purposes of claim construction, the issues coincide with one another. So the '544 patent is really our reference.

The patent deals with the issue of malicious e-mail attachments. And let me provide a little background which will help put the patent, the invention, and the claim construction issues in some context.

So in the early 2000's and late 1990's, e-mail proliferated. And with the proliferation of e-mail it allowed for the ease of transmission of e-mail attachments that contained programs. So not just any attachment to an e-mail, not a document or a piece of text, but an executable, a program that could be attached to an e-mail. And this patent deals with the issue of executable attachments. So I put on the screen two examples of executable e-mail attachments that were viruses. One example was from the May of 2000 time frame, the ILOVEYOU virus. The subject line said, "I love you," there was an attachment, and if you clicked on it, the program did some harm to your computer, might have erased some files, and then as if that wasn't enough, it replicated itself and sent itself out to everyone in your contact list. So it caused a lot of havoc.

Another example of the attachment called message.zip, and it was this MyDoom virus, and it replicated and sent to a particular server which could overwhelm a server and cause a denial of service attack. So these could cause significant problems for yourself personally as well as the community at large. How did the prior art detect these viruses? Generally, you waited to find one, and once you saw it you developed a signature and you said, you created the fingerprint. You said, "That's the ILOVEYOU virus." And so when it showed up again you could detect it. But what if it was different? If it was different, you would have trouble picking it up. The signatures wouldn't work because you hadn't seen it before. And so the inventors in the Columbia labs were working on applying their techniques in machine learning to detect malicious e-mail attachments.

Now, the background of the '544 patent sets this up, which says, "The invention relates to systems and methods for detecting malicious executable programs, and more particularly to the use of data mining techniques to detect such malicious executables." So we are talking about the use of machine learning techniques to detect things that you hadn't seen before.

Let me provide a little context for machine learning in the context of these patents. So the first general principle is you want to collect information about the executable. You want to look at the features of the file without running it. You want to collect features about the file so you can build a rule that you can then apply the features in the new file to to determine whether it is malicious or benign. Even though you haven't seen it before, you build a model based on features of a training dataset that you have seen before, you see the new file, and then you run it through your model to determine whether it is malicious or benign.

So you have the information collection stage, the rule set development stage, and then ultimately the comparison to the rule set.

The principles of the work reduced to several key concepts that are reflected in the patent disclosure and in the claims. So the first step that's described is you filter the e-mail attachment. Then you extract byte sequence features. The features that we are talking about being collected from this large set of data and from the files that are being inspected are referred to in the patents as byte sequence features.

The next thing you do is you build your rule set based on the features that you have collected. And then you classify the file as malicious or benign by comparing the features to the rule set.

Now, these are the general principles that, and I'll repeat them because they are set out specifically in the claims, and it will highlight the specific claim construction issues that are before the Court. On Slide 22, you can see an example claim, which is from Claim 1 of the '544 patent. And the claim relates to a method for classifying an executable attachment. So we are talking about executables that are attached to an e-mail. Step A, you filter said executable attachment. There is no claim construction issue here with regard to the filtering step. Element B: Extracting a byte sequence feature from said executable attachment. Byte sequence feature is at the heart of the dispute for the parties today, so this is, Element B is a key claim element we will be talking about in more detail. Element C is broken up really into two parts, where after you have done the filtering and extracting, you classify the executable attachment by comparing the byte sequence feature to the model that you have created using your machine learning techniques.

Then the claim in its last clause contains another element that goes with Element B, which is that the byte sequence feature that gets extracted, for purposes of Claim 1, cannot be anything, it needs to include a byte string representative of resources referenced by said executable attachment. In general, what that is referring to is you can extract features from the file, but in Claim 1, you need to at least include a byte string representative of resources that are referenced by the file. "Resources" refers generally to operations that the file may perform, resources that the file may call on in a system to perform, such as if a file has a link to another routine that's in the operating system, it is sometimes referred to as a Dynamic Link Library or DLL. You might see in your computer in an attachment, this is .dll and you are wondering what that might be. A DLL is a call to another library in the operating system. So that might be an example of a resource that's referenced by the executable. Claim 1 talks about extracting the byte sequence feature, and it requires this last component that it include a byte string representative of resources.

We are going to talk about in more detail about those points in the context of the claim construction disputes.

There are three basic disputes. The first two relate to the byte sequence feature and this byte string representative of resources element. We are going to address those two together. The parties have agreed to address them together, because they are intertwined. Symantec may address them in another order than I do, but we are going to address the wherein clause, the representative of resources point first, because the arguments on that element inform the understanding of what a byte sequence feature is.

So let me highlight what the parties' competing positions are on this point. So Columbia's construction of the wherein clause is simply to confirm what the claim says: That the byte sequence feature includes a byte string representative of resources referenced by the executable attachment. The claim is talking about extracting a byte sequence feature, and it is talking about including at least a certain kind of byte sequence feature, one that includes a byte string representative of resources. That's Columbia's construction of this term.

Symantec's position is that the claim is indefinite. Now, it is not your usual indefiniteness argument. I'm not entirely convinced it is an indefiniteness argument. But Symantec's position is not that there is a claim term such as that the claim requires something to be blue, and you are not really sure if it is blue because people might have different understandings of what blue is. That's not their argument. The argument is that the claim is internally inconsistent because it claims two different embodiments. That the byte sequence feature in Element B is mutually distinct and mutually exclusive from what I am referring to as Element D, this representative of resources element. And that the claim, therefore, is internally inconsistent, and therefore, should be indefinite.

Symantec, in making this argument, has gone off track right from the start. When you read the entirety of the disclosure, it is clear from the way the invention is described from the beginning to the end that a byte sequence feature includes as an example a byte string representative of resources referenced by the executable. That this is an example of one of the kinds of byte sequence features that you can extract from the file.

Let me try to explain it graphically. The point of the invention is that you create a dossier about the file. You collect the features that might be pertinent to your analysis to determine whether it is going to be malicious or benign. And I've displayed a folder called "Byte Sequence Features." And the byte sequence feature folder which you can pull out of the file can include the instructions the file might perform. It might include more particular information, such as resource information, the DLL that I mentioned earlier as an example. Or it can include plain text from the file. So there are different parts of a file, and you can include any of those pieces of information can be extracted from the file and they are byte sequence features. They are described this way in the patent.

So you can create this dossier in the file, they are all byte sequence features and examples of byte sequence features. Now, Symantec's position starts from an incorrect premise, which is that byte sequence features are only one kind of feature, and that everything else is something completely different and that when the patent mixes the concept, it has gone astray and there is something wrong with the claim. That's completely inconsistent with the disclosure. So you see on the screen, Symantec puts each embodiment in a different box, byte sequence features, which it says are only a certain kind of information, it is only the machine code, that's Symantec's construction. There is resource information. That's entirely separate, not a byte sequence feature according to Symantec. That's wrong. Encoded string features. They are saying that's not a byte sequence feature. That's wrong. So Symantec, this is the entirety of their -- the entire basis for their argument and it is flawed from the start.

Let me explain the patent, how it describes byte sequence features. You can start from the Abstract, the beginning of the patent. It says, "Byte sequence features are extracted from the executable." That is a core of the patent. It is a core principle of the patent. And it is developed step by step when you go through the disclosure. So moving from the Abstract, which lays out this basic principle, I go to the Summary. The Summary of the Invention explains how the byte sequence features that are extracted comprise extracting static properties of the executable. The point of the '544 patent is I don't have to run the file. I'm going to put a magnifying glass over the file contents and extract static properties from the file and those static properties in the patent are referred to as the byte sequence features.

The Summary also gives examples of extracting the byte sequence features. I can extract the entirety of the bytes in the file. I can take the file and I can convert the executable attachment from binary format, 1's and 0's, to hexadecimal format, which is a base 16 format, and a different set of -- different way to represent the 1's and 0's. But it is just translating the 1's and 0's into hexadecimal format. I can do that for the entirety of the file. That's one option. Or I don't have to extract the entire file. I can extract a byte string representative of resources. I can just extract DLL information, as an example. That's another embodiment. So the Summary of the Invention gives the basic explanation. I extract the static properties just on -- going back one, Slide 32, extract static properties, and then continuing on, I can convert the file to hexadecimal, the whole file, or I can extract a byte string representative of resources. These are examples. Symantec is critical of this argument by saying the Summary just references the claims, just repeats the claims and should be given no weight. Number one, as a legal matter, the cases say that there is no reason that the Summary of the Invention does not get as much weight as the rest of the disclosure. Number two, it is incorrect. The Summary is more than the claims. It provides the context for explaining what the invention is.

Now, if I continue from the Summary, the detailed description builds this out further, this extraction process, extracting the features. The extracting of the features are from, referred to as Step 20 in the patent. It says, "The next step of the method is to extract features," and it is referred to as Step 20. That begins the discussion from Column 5, at the bottom of Column 5 of the patent. And when you look at the patent itself, this extracting the features discussion goes from the bottom of Column 5, Line 57, and it goes all the way through Column 6 and through the end of Column 7. And that's describing this feature extraction in much more detail. So it is Step 20 and it says: "Features in a data mining framework are defined as properties extracted from each example program in the dataset, e.g., byte sequences." So it is referring to the features that are extracted as byte sequences. This is -- the entirety of the disclosure is referring to the extraction of byte sequence features.

Let me continue because -- with the detail that's in the specification. So Figure 1 is just a graphic illustration of this Step 20, which is the extraction process. And then the text says that there are lots of ways you can do this Step 20, you can do this feature extraction. And it describes the kinds of things that you can extract. So let me continue just to walk through the detailed description, how it matches with the Summary of the Invention. So the Summary of the Invention gives us one option that you can convert the executable attachment from binary to hex. You can do this extraction of the entire file. And in Column 6, this is described in more detail. You can do this by using a known utility called hexdump, and hexdump will extract all the contents of the file and output them into hexadecimal strings. So you can extract the entirety of the file. This is one embodiment. And then, so that's going to include everything. It is going to include the instructions, the resource information, plain text, the entirety of the file. And the patent talks about why there are some advantages of that.

Now, this is one example of byte sequence features, the entirety of the file, of course. Then another example from the summary mapped to the disclosure is you can create a byte string representative of resources that are referenced by the executable. And then if you continue in Column 6, it says, "Additional methods of feature extractions are also useful to carry out this same Step 20." I don't have to extract the entirety of the file. I can extract a part of it. And I can extract resource information from the binary that provides insight to its behavior. But the point of this disclosure is that I can extract the entirety of the file, or I can extract resource information, and the resource information is a kind of byte sequence feature. It is an example. It is not an entirely separate animal that has nothing to do with the byte sequence features that are extracted from the file. The disclosure refers to the extraction of all the bytes in the binary, all the bytes in the file, or you can extract a portion of it. And it discusses the extraction of resource information.

An example of resource information is depicted in Figure 3, which it says if I just extract resource information as the byte sequence feature, this is what it might look like, if I pull out the dll's and I can create a string that captures just the dll's, it might look like something in Figure 3. And all Figure 3 really is is, it is a conversion of the bytes that are in the file representing this information. I've translated the 1's and 0's into text that I now can understand, and I can represent the bytes as ASCII text, so numbers and letters, and create a string that represents the dll's. So this is an example in Figure 3 of a byte sequence feature that is representative of resources. It is one example provided by the patent.

This concept of byte sequence features being an example of -- I'm sorry -- this concept of a byte string representative of resources referenced by the executable as being an example of a byte sequence feature is not something that claim out of the blue and is not some strange creature. It has been in the application process from the beginning. The original application included claims that were to the same effect. The originally filed Claim 1 was you extract byte sequence feature, and then you have, there's dependent, originally filed dependent claim, wherein you extract the particular kind of byte sequence feature, this byte string representative of resources. So the inventors have regarded the resource information as an example since the beginning.

One argument I'd like to address preemptively is Symantec in their briefs argue that Claim 28 shows that byte sequence features and resource information are different things. And if you look at Claim 28, this is a system claim, Claim 1 is a method claim, and so Claim 28 requires a feature extractor. And the feature extractor requires you to extract a byte sequence feature, and it also requires that it is further configured to create a byte string representative of resources. Symantec says, "Well, those are two completely different things. I don't have a problem with this claim, because they are two different things and it is claiming them separately." And I would disagree. As we explained a moment before, the byte string representative of resources is an example of a byte sequence feature; the feature extractor in this claim simply needs to be configured to be able to do that, to extract -- to create a byte string representative of resources. But more to the point on why Symantec is wrong is, if a byte sequence feature does not include resource information, then Step C, which is what you do with the byte sequence features, would do nothing with this information that's representative of resources. So you would extract the byte sequence feature, you would be configured to create a byte string representative of resources, and then when you are doing the meat of the comparison to determine whether you are malicious or not, you would do nothing with this resource information, because all the claim says is you compare said byte sequence feature to the rule set.

So if byte sequence feature doesn't include resource information, then this claim doesn't make sense under Symantec's view. And it all stems from Symantec's misinterpretation of the entirety of the patent. When Symantec puts byte sequence feature in one box, resource information in another box, and print the string information, other types of information, in another box, they have gone off track. And the byte sequence feature is the dossier about the file, and the instructions, the resource information, and plain text information, those are all examples of byte sequence features.

THE COURT: Let me interrupt you for just a second. I have a jury out and I'm checking to see if they are here.

MR. BIRNHOLZ: I completely understand, Your Honor.

THE COURT: All right, we will go down for a few minutes and get set up in my other case. All I have to do is bring the jury in and send them back out and then we will get back to you all. So we will take a break, get the McDonnell folks in here.

(Recess taken from 9:50 a.m. to 10:01 a.m.)

THE COURT: All right. What happened to our audience? I guess nobody is interested in the patents.

MR. BIRNHOLZ: I lost all my fans.

THE COURT: All right. Go ahead.

MR. BIRNHOLZ: Thank you, Your Honor. To bring us back to this discussion, I wanted to highlight what was on Slide 29 and to wrap up this particular section. The indefiniteness argument as far as this resources referenced by the attachment. It fails from the starting gate because of a fundamental misconception of what a byte sequence feature is. And I think this is a useful image to keep in mind, that the byte sequence feature is the dossier of the file, and it can contain a variety of kinds of information, and it can contain instructions from the file, resource information about the resources that are referenced by the file, plain text information that can also provide very useful information about the file's behavior. Those are all examples of byte sequence feature.

And so with that context, our construction reflects that the last element, this is Slide 41 at the end of this particular section, is "wherein the byte sequence feature includes a byte string representative of resources referenced by the executable attachment." I think that it is consistent with the claim language, it is consistent with the disclosure, the description of the invention, and there is no indefiniteness problem.

Let me move to the next element, which is related, which is actually the parties' competing position on what a byte sequence feature, how that should be construed. And it is really -- the reason that we are having this dispute is because of a limitation that Symantec wishes to read into the claims and to limit that to a particular kind of example of byte sequence feature. And let me go right into that.

So Claim 1 again, we are talking about Element B for this particular issue, "Extracting a byte sequence feature." The parties' positions on this are at Slide 44, which is, Columbia provides a construction of byte sequence feature as "a property or attribute of a sequence of bytes which may take on a set of values." And the parties are similar in terms of what a feature is, but we haven't defined "feature" separately like Symantec does. Symantec says that's "a property or attribute of data, which may take on a set of values." Then they provide a separate construction of byte sequence feature which they then use "feature" again. It is "A representation of machine code instructions of the executable."

So Symantec has limited byte sequence feature to "only a representation of machine code instructions." So they are reading in that particular example to the claims. They are trying to import that particular example and limit the claims to that particular embodiment. And let me describe some attributes of a file generally which might help put this in further context as well. So first, as a general matter, a file can have different components. It can have a header portion, it can have data, and it can have instructions. And the image is meant to depict a typical format for what's called a portable executable format file. It is an example of an executable. And there are specific sections of the file.

But when you step back from what's in the file, there are different components, the general components: header, data, and instructions. And there is really no dispute about this particular point, that files contain different components. And the experts have submitted Declarations on this issue which, at the next slide, which explain that the Windows PE files include a header that contains information about the file, such as the file size, the names of dynamically-linked libraries, those DLL's that I talked about which can call other functions, and this is Symantec's expert. So this is not an area of dispute. And then Columbia's expert, Professor Szajda, says that a file can contain the instructions that are performed by the CPU, but it is not the only thing that's in a file. You can have DLL information, which are an example of resource information. You can have plain text.

The point is that there are different parts of a file. And this is important for the claim construction because when you go to the claim language, where it says "extracting a byte sequence feature."

So my first point on the construction is the claims are not limited to machine code instructions. This just says "byte sequence feature." So number one, there is no machine code limitation in the claim language itself. And then when we go to the specification, we also see that the specification does not limit a byte sequence feature to machine code instructions. Part of that I explained earlier. But let me focus on this particular point in the specification from Column 5, which says "a feature is a property or attribute of data such as byte sequence feature which may take on a set of values." So Columbia's construction is drawn right from the specification at Column 5, Lines 57 to 64.

If the Court is unsure as to what "which may take on a set of values" means, that's going to be the attributes that are specific to the particular feature, such as if you looking at an apple, how thick the skin is, how much juice comes out, how round is it. So the values are just the particulars of the properties or attributes. So Columbia has drawn its language right from the specification. It is clear from the specification that byte sequence features may be extracted from all or a part of the file. Now, Symantec is trying to limit the construction to only machine code instructions. And again, I'll quickly walk through the specification on this point that makes this clear: "that the byte sequence features may be extracted from all or a portion of the executable."

In the Objects of the Invention that are described in the Summary in Column 3, one object is to "provide a data mining technique which examines the entire file, rather than a portion of the file, such as a header." We talked about that earlier. You can examine the entire file or you can examine a portion. And the patent describes ways of examining the entire file and ways of examining just a portion. They are all extracting these byte sequence features from the file.

Then when you go into the detail in the Specification, Column 6, where there is this discussion of the extraction process and the options that are available to extract byte sequence features, one is you can examine a subset of the data. So this is describing some methods where you can extract the particular resource information alone or you can extract information about the entire file. So Option 1 is the entire file, Option 2 is a portion of the file. And this is addressed at Column 6, Lines 2 to 23. And then it continues with another example of portions of the file. Well, you might want to just extract what are called plain text headers. Now those are also byte sequence features. And these are the features that can be extracted from files that are not in that Windows portable executable format, so you can apply this method to a variety of different kinds of files. So you can extract the plain text headers, and the plain text headers might include resource information as well. But they can include a variety of information about the file. So all or portion of the file can constitute the byte sequence features.

Where does Symantec get its construction from, what are they relying on? There is a paragraph in the Specification which describes one exemplary embodiment, which is this concept of looking at the entire file as your byte sequence features. And it is at Column 6, Lines 7 to 22, and I've reproduced it here on Slide 53, which is in the exemplary embodiment, hexdump was used in the feature extraction step. Hexdump is this utility that can extract all the contents of the file and output them in these hexadecimal strings which you can then analyze. Symantec says -- well, let me read what this says. I'm going to rely on this sentence: "The byte sequence feature is informative because it represents the machine code in an executable." That one line is the linchpin for Symantec's argument. They say, "Okay, it says it is informative because it represents the machine code." Therefore, then, Symantec makes the incorrect leap to, "Well, a byte sequence feature has to represent machine code instructions, and only machine code instructions."

Well, that's just completely wrong. The paragraph itself says it is informative because it represents the machine code. Okay. So it represents the machine code. But it doesn't anywhere say it represents only the machine code instructions. In fact, we know this from Professor Szajda's Declaration and from it being well known in the art about what hexdump does, this hexdump utility. It extracts the entire file, including machine code instructions and the other components of the file. And I think in the second line on this slide, it says, "Hexdump as is known in the art." So hexdump is not a creature of the invention; it is a known utility you can use to extract the entire file.

And the specification itself further explains this in Column 13, when we are talking about this process in more detail, when you are actually performing the classification by looking at these new files. You can transform the binary files in the attachment into a byte sequence of hexadecimal characters. It is referred to as transforming the entire binary into hex characters. It says, "This approach involves analyzing the entire binary, the entire file, rather than portions such as headers." So you can look at the entire file, which includes the machine code instructions and everything else, or you can look at a portion. So there is nowhere in the specification that limits what a byte sequence feature is to machine code instructions. Symantec is merely reading that in.

One argument that we saw in the briefs and you may hear about today is that all of this should be overridden by what's in the provisional application, and that the discussion in the patent really doesn't mean what it says, because of their interpretation of the provisional application. And the provisional application in this case was a paper, a research paper that the inventors worked on. They published the paper and also filed it as the provisional and then the full application was developed. The provisional is entirely consistent with the position that I have articulated; that you have the options of looking at the entire file or portions of the file. And you look, one of the sections in the provisional which describes this feature extraction talks about how you can use a certain utility and extract some particular DLL-type information, or you can extract strings. But then the last section in the section is called "Byte sequences using hexdump." So we have talked, the provisional provides examples of these byte sequence features. And then it says, "You can also use hexdump," which we have explained is a tool that transforms the file into hexadecimal files. And it says, "Analyzing the entire binary gives more information because you have extracted the entire file." And it is another option to extract byte sequence features. You can extract the particular resource information or a string or you can use hexdump and use the entirety of the file.

And so the provisional is consistent with our position, and it is also consistent with the exemplary embodiment in the specification. And when you go back to the specification, at the end of the paragraph that Symantec relies on about this hexdump embodiment and the machine code instruction, the machine code reference, that paragraph itself ends: "Each byte sequence in the program is used as a feature." And that's consistent with the concept that hexdump outputs the entire file. So for Symantec to argue that byte sequence feature is limited to just machine code instructions is inconsistent with the entirety of the disclosure, including the provisional.

Now, the last point on this section is Symantec is going to say, well, and I think Symantec's counsel said this in their introduction, in their introductory remarks, "We just want to claim a feature." And the essence of that argument is, "We are just rendering the words byte sequence superfluous. We are reading those out of the claim." That is also incorrect. When you look at Columbia's construction, it says "a property or attribute of a sequence of bytes, which may take on a set of values." And when you read this in the context of the claim, it must be based on a sequence of bytes from the executable. It is not extraneous information, like who sent the e-mail that had the attachment, who received the e-mail that had the attachment, who received the e-mail that had the attachment, what time was the attachment sent, how many copies of the e-mail did you receive? That information would not be derived from the executable. And so the construction is consistent with the disclosure. It does not read in any improper limitations, doesn't read out anything from the disclosure.

And so I'll close this section by saying that the byte sequence features are not limited to machine code instructions, and the structure of the claim has integrity and is faithful to the disclosure and is not indefinite.

Thank you, Your Honor.

THE COURT: All right.

MR. RUDIS: Good morning, Your Honor. Alex

Rudis on behalf of Symantec. I'll pass these up. So Your Honor, we proposed constructions for "feature" and "byte sequence feature," because they are two different terms. And "feature" is the more generic of the two terms, which is why we construed it as a property or attribute of data which may take on a set of values. As counsel said, I don't think we actually disagree on the general definition of "feature." That construction comes straight out of the specification. So the patent does actually describe three different methods of extracting features. One of those methods extracts byte sequence features, which the patents very clearly describe as a representation of machine code instructions.

The other two types of features they don't. They extract either resource information or encoded strings. And we will get into some of the slides that counsel showed you in particular, the slide with the folder that said "byte sequence feature" and the three different things within it were "instructions," I think the second one was "encoded strings," and then "resource information." Our problem with their construction is, they have just replaced "byte sequence feature" -- or "feature" with "byte sequence feature." My point is, "feature" is the generic term. So you have feature, and that's as we have construed it, "a property or attribute of data which may take on a set of values," and then there are types of features. One is a byte string feature, which is what is in the claims and what is the dispute here. And then there are other types of features. And in that folder, the other types of features was the resource information. That's a different type of feature, and we will see that in the specification. And then the third one is an encoded string, which is a different type of feature. I'm going to pick up where counsel left off, which is starting to talk about the byte sequence feature construction, and then come back to the indefiniteness.

Let's first talk about our construction for byte sequence feature. Go to Slide 6, please. So you saw, Your Honor saw this Figure 1 in counsel's slides, and we think it is a good starting point. If you see Box 20, "Extract features from data." And that's "features," the general features, right? So we see in the highlighting here "A feature is a property or attribute of data which may take on a set of values." So the parenthetical is, "such as byte sequence feature." Counsel pointed that out to you. What they want to say is, wherever you see "feature," it is a "byte sequence feature." But no, "such as byte sequence feature." If we look up a little bit, the second sentence, "Features in a data mining framework are defined as properties extracted from each example program in the dataset, e.g., byte sequences." So "For example, byte sequences." Not "Features are byte sequences," but "One example of features are byte sequence features." So this permeates both parties' briefs, quite frankly, and the dispute is, well, are byte sequence features features, or are byte sequence features one type of feature and that you have these other types of features, which quite frankly aren't claimed, and if they are in the same wherein clause, it is indefinite, because they are different.

So I don't think either party really believes that there is a plain and ordinary meaning for byte sequence feature, so we need to go into the specification to see where we would find support for a construction.

Let's go to Slide 7. And this is what counsel pointed out to you, which is part of the basis for our construction, what we will see here, the patent says: "The byte sequence feature is informative because it represents the machine code in an executable."

So we heard a lot about hexdumps. And what hexdumps does basically is it converts a binary file into a hexadecimal file. And it does that to the entire file. So you have some binary, and then you have hexadecimal. So what happens then? After the hexdumps are created, features are produced in the form illustrated in Figure 2 in which each line represents a short sequence of machine code instructions. And then later, down the last sentence here, which isn't highlighted, I'm sorry, "Each byte sequence in the program is used as a feature." Counsel pointed that out as well, to make the point, "Well, it is the entire file, so it can't just be the machine instructions." But that's what we are saying. We don't really dispute that hexdump creates conversion of the binary to the hexadecimal of the entire file. What happens is, after that's created, after hexdumps are created, features are produced in the form of Figure 2 in which each line represents a short sequence of machine instructions. Those are the byte sequence features.

And the patents tell us precisely why byte sequence features are helpful. Let's go to the next slide. This is just the same portion of the specification, but what it says is, "In the analysis, a guiding assumption is made that similar instructions were present in malicious executables that differentiated them from benign programs, and the class of benign programs had similar byte code that differentiated them from the malicious executables." So what this is saying is, the machine code instructions, in whatever format, but in this format they are hexadecimal, the ones we have chosen as byte sequence features, those are the most helpful for us.

There are other portions of the file that you have in hexdump or binary, the portable executable file that counsel showed you. Yes, sure, there are other things in the file. But what's most helpful for you are the machine code instructions.

So you take these byte string sequences, or you take these byte sequence features, which represent the machine code instructions, and those are the most helpful for you. It says it right here.

So then the patent goes on to talk about other types of features that you can extract. So again, when counsel was explaining these other things like resource information and encoded strings, he was swapping the words "feature" and "byte sequence feature" to say, "Well, these are other examples of byte sequence features." But no, these are other examples of features. "Many additional methods of feature extraction are also useful." According to another approach to feature extraction is to extract resource information, right? So remember counsel's Slide 29, which was the folder, right? So really, it is features, and one is instructions, which is byte sequence feature. The next one, "Many additional methods of feature extraction is to extract resource information." So that's the second one. That's a separate type of feature, and that's what we are looking at here.

So let's go to the next slide. So the patent also, it goes on to talk about how these are alternative methods of feature extraction. And the first portion of this is now talking about byte sequence feature, and then juxtaposing that to the other types. "This byte sequence is useful because it represents the machine code of an executable." That is byte sequence feature. It goes on to say, "It is understood that the feature extraction step," not byte sequence feature extraction step, "herein is alternatively performed with a binary profiling method in another embodiment as described above and illustrated in Figures 3 and 4." And that is a byte string representative of resources. That's the resource information, what we are talking about here in the bottom of the column.

So the third type of feature, or method of feature extraction in this patent, is encoded string. So it is plain text. That's not really in dispute here. It is another -- it might be disputed that that is a type of byte sequence feature or not from Columbia's perspective, but from our perspective it is just a third type of feature that's disclosed in the patent. And that, again, looking at Columbia's Slide 29, that's the plain text. So this would be features, not byte sequence features. And in each folder, each little file in that folder would be byte sequence features which aren't machine code instructions or representations of machine code instructions. The next one would be resource information, and the third would be plain text.

So we heard a little bit about the provisional application to these '544 and '907 patents. And we don't, actually, I don't think we ever said that they were any different. They are entirely consistent from one another. So let's go to Slide 11. So the first point, and it is an important point, is that this '622 application, which is one of the provisional applications to the '544 and the '907 patent, that was incorporated in its entirety, incorporated by reference in its entirety into these patents. Right? And Your Honor might see some cases cited by Columbia in their responsive brief talking about, "Well, it is not the provisional that matters, it is the as-filed specification." In fact, the patent in that case that Columbia cited, that wasn't actually incorporated by reference. So that's distinguishable right there. But the point is, here, we have the '622 application, which was incorporated by reference in its entirety into these patents. That makes it part of the intrinsic record. We don't think the two are different as it relates to what we are talking about here, the claim construction issues for byte sequence feature.

So let's go to the next slide. And there's a lot of information on this slide, but really this is just to show that the description, I'm looking at Slide 12 now, the description of how resource information is extracted from the provisional on the left of the slide is the same as it is in the specification. For the next slide, again, for the description in the provisional application, the '622 provisional about using hexdump to get byte sequence features, which are machine code instructions, is the same as it is in the specification. So there isn't much difference between the two.

And I'm not sure if we have made that statement, but as we can see, the provisional says the same thing as the '544 application says as it relates to byte sequence feature resource information, which is, they are different features, different types of features.

Let's go to Slide 14. So this is what the provisional application, the '622 provisional application says about feature extraction. And again, this is part of the intrinsic record, because it was incorporated in its entirety into the '544 and the '907 patents. So first it says: "We statically extracted different features," not "different byte sequence features," "We statically extracted different features that represented different information contained within each binary." Then the next portion of the '622 provisional application says: "The byte sequence feature is the most informative because it represents the machine code in an executable instead of resource information." So resource information isn't a type of byte sequence feature, it is a different type of feature.

So the '622 application, consistent with Symantec's claim construction, it clearly defines byte sequence feature as representations of machine code instructions, and it contrasts byte sequence features with resource information.

That sort of segues us to the indefiniteness portion of this, which -- well, let me, before I go there, the main problem with Columbia's construction, again, is that everywhere they see "feature" in the specification, they want to say, "Well, it actually means byte sequence feature." Right? And what they say is, "Well, we have changed the construction a little bit because we added sequence of bytes." But every program ultimately boils down to a sequence of bytes. Right? Professor Szajda's second Declaration, that's to their responsive brief, if you look at it, he basically says, "Well, there are many ways to display sequences of bytes and any sequence of bytes is a byte sequence feature. You know, some of them could be machine code instructions, some of them could be a portion of the PE header, some of them can be in plain text." But, "Hey, anything can be a sequence of bytes in a program." So you follow that logic, if anything in a program, and in this case a potentially malicious executable, can be boiled down to a sequence of bytes, well, then anything can be a byte sequence feature. But that's not what we have here. We have "features," and we have "byte sequence features" which were fairly explicitly defined in the specification, and you have other types of information that could be a feature.

So we start with the construction for byte sequence feature, because we believe it informs why this wherein clause is indefinite. So let's go to Slide 15. Now, we have already seen this, but just to highlight, the '544 and the '907 patents are very clear and very explicit that these are additional methods of feature extraction. One is byte sequence feature, another is resource information, and another is encoded strings. Let's go to the next slide. And so we have seen the provisional application, we have seen this slide before, right? We statically extracted different features that represented different information. And again, in this second quote, "The byte sequence feature is the most informative because it represents the machine code in an executable instead of resource information."

So let's go to the next slide. So this slide in the bottom wherein clause is what we are asserting is indefinite, is you have these two different types of features. "Byte sequence features" and "Byte string representative of resources" referenced in the same clause here. And it is nonsensical. So we had, I think in the beginning of the tutorial, we had, well, you are looking for -- from Columbia's tutorial, you are looking for fruit and you can look for different -- so that's feature, the general class. But then you are looking for types of features, so you have, I think it was seeds, does it come from a tree, what color is it, right? So, well, this is saying, well, you have -- you are looking for a fruit and the color also includes the seed, wherein the color is a seed. That's what this is saying according to that example.

Or another, well, you have mammals, right, that's a type of feature, the general class. And then you have types of mammals. You have humans and you have dolphins. This clause here is saying, wherein a human, wherein one type of human happens to be a dolphin. These two things are sub-classes of feature, but somehow here, resources happens to be a sub-class of byte sequence feature. And it is nowhere in the specification.

Now, counsel pointed you to the Summary of the

Invention, and then everywhere else in the detailed description of the embodiment, you don't find anywhere where you see "resources" meaning it is a type of byte sequence feature. "Resources," or "Representative of resources," might be a type of feature, but nowhere in the detailed description of the embodiment or of the invention here do you see those two things meaning, well, you could have resources that are part of a byte sequence feature. It is only in the Summary of the Invention. And you can dismiss the Summary of the Invention where it parrots the claim language verbatim. We have seen cases that say that, and it is cited in our brief. So if their only evidence for saying, "Well, resources are a type of byte sequence feature" is in the Summary of the Invention and all the other evidence that we have shown you that says they are different, they are different, they are different in the actual detailed description of the invention, well, we think the evidence weighs in favor of saying that they are different. And that's both in the specification and the intrinsic record, which would be the '622 provisional application, which was incorporated by reference in its entirety.

So I think with that, we can probably move on.

THE COURT: All right.

MR. BIRNHOLZ: Brief rebuttal, Your Honor?

THE COURT: Sure. Go ahead.

MR. BIRNHOLZ: Thank you. A couple of brief points because I know we have a lot to cover today. So first, once you run this hexdump utility and convert the file into hexadecimal format, there is no way of determining what are the machine code instructions and what are the other parts of the file. The hexdump just dumps everything in the file. So opposing counsel's argument is really a mischaracterization of that one passage in the specification. And when you look at Column 6, Lines 31 to 32, it repeats, it says, "Each byte sequence in the --" "Each byte sequence in the executable is used as a feature." And so that line itself completely undermines the argument that a byte sequence feature is limited to just machine code instructions.

The points, if I could bring up Slide 72. So counsel said how strings, that's a completely separate embodiment, he made that point. And that's part of the argument how you have these separate boxes that they are making. You've got strings and resources and byte sequences and they are all their own boxes unrelated to each other. Table 1 in the patent is a list of strings that can be executed, that can be extracted from a file. And when you look at the strings that can actually be pulled out of a file that are listed in Table 1, there are examples of things that are resources. So this advapi that's listed in Table 1, that's also the name of a DLL, which is a resource. There are text strings that say "Create File A" or "Write File." Those are resources. So they are examples of instructions, resource information, and plain text are all examples of byte sequence features. And when counsel said "Every program boils down to a sequence of bytes," I would absolutely agree with that. And those sequence of bytes can be represented in different ways. They can be represented as instructions, as resource information, as strings, and they are all examples of byte sequence features.

And that is clear from the Summary of the Invention, which was not in the provisional application, the detailed description as well, and the provisional itself. When counsel said, "Well, the provisional is distinguishing byte sequence features from the other information," again, it says that the -- it said, "The byte sequence feature is the most informative because it represents the machine code in an executable instead of resource information like libBFD features." So all that sentence is saying is that the byte sequences and the byte sequence features that are extracted from hexdump, which is the entirety of the file, is the most informative because it represents the machine code instead of only resource information. Hexdump is the entirety of the file. There are other examples in the provisional that are parts of the file. And the provisional is consistent with the disclosure. And the Summary of the Invention specifically describes resource -- "byte strings representative of resources" as an example of the byte sequence features that can be extracted. It is the words "byte sequence feature" and "byte string representative of resources" are used together to describe that embodiment in the Summary of the Invention, and it is explained in great detail in the detailed description.

So with that, I think I would urge the Court to adopt our construction that the machine code instructions are nowhere to be found in the construction, and should not be read into the construction of byte sequence feature. It is a property or attribute of a sequence of bytes, which may take on a set of values, and the claim is logical and makes perfect sense in light of the disclosure. Thank you, Your Honor.

THE COURT: All right.

MR. RUDIS: May I?

THE COURT: Go ahead, extremely brief.

MR. RUDIS: So the only thing I wanted to add was, counsel said, well, the Summary of the Invention wasn't in the provisional. Our position is the Summary of the Invention just parrots the claim language. So it makes sense that the Summary of the Invention wasn't in the provisional because the provisional actually didn't have any claims. So there was nothing to parrot. That's all I wanted to add.

THE COURT: All right.

MR. BIRNHOLZ: No surrebuttal or sur-surrebuttal.

THE COURT: Let's move on.

MR. BIRNHOLZ: I realized we have another term. So the last term in this is "e-mail interface." We believe that the parties' constructions of "e-mail interface" reveal a pretty fundamental dispute that we think is easy to resolve in our favor. Columbia's construction of "e-mail interface" is hardware or software that interacts with e-mail traffic and other e-mail processing components. It is a definition that is consistent with the words "e-mail interface" and how it is described in the patent.

Symantec's position, you can see, reads in this requirement that the component that reintegrates filtered e-mail back into normal traffic, that it has to do this reintegration function. Let me explain why this is incorrect. So first, "Interface." "Interface" is a term that's used to refer to something that communicates between two things. In the dictionary, you look up interface, "Some form of electronic device that enables one piece of gear to communicate with another or control another." "Interface" is something that enables communication.

In the context of an "e-mail interface," how might that be understood? It is something that's going to sit between e-mail traffic and components that will process that e-mail traffic. And that's how the term e-mail interface is used in the patent. When you look at the patent specification, Figure 9, Figure 9 in Box 232, which I've highlighted, is "an e-mail interface that sits between e-mail traffic and the rest of the processing components." And the arrows that go back and forth show multiple functions.

The specification describes the different functions that are possible from the e-mail interface. It can reintegrate filtered e-mails, it can send the model generator, each attachment, it can add warnings to the e-mail, it can quarantine the e-mail, or send copies of attachments to the filter interface. So these are things I've highlighted on Slide 65 that are examples of what the e-mail interface can do.

Now, Symantec's construction just says it has to reintegrate filtered e-mail back into normal traffic. It is clear that the e-mail interface is not limited to only reintegration. Sure, that's one function it can perform, but that's not the only function it must perform, because here is an example. You can quarantine the e-mail after you analyze it. That's the opposite of reintegrating. So Symantec's construction would be improperly limited to one embodiment. And quarantining is the opposite of reintegrating.

Symantec says that, "Well, there are other functions that can be performed" and that they need to be set out in the claims and they need to be set out in the construction. Well, the claims define additional functions that the e-mail interface provides, Claims 32, 41, and 42 are examples of additional functions that are spelled out for the e-mail interface. And it would be inappropriate to read in this reintegration limitation to the claim. So Columbia's construction is consistent with the ordinary meaning of the term and the disclosure, that it is "hardware or software that interacts with e-mail traffic and other e-mail processing components." And we would urge the Court to adopt that construction and reject Symantec's much narrower construction limited to only a particular embodiment.

THE COURT: All right. Thank you. Symantec?

MR. BIRNHOLZ: If I may, I don't know if we gave a set of our slides to the Clerk.

THE COURT: Sure, go ahead.

MR. RUDIS: Let's go to 19. I'll be brief with this one, Your Honor. Columbia doesn't actually use anything in the intrinsic record to support its construction. All they are using is a dictionary definition and that's simply for "interface." Our construction, Symantec's construction comes straight from the specification. And it is the only thing in the specification that is actually done by the e-mail interface. Maybe it does other things, maybe it could do other things. The specification is actually pretty clear that it may do things. But this, the component that reintegrates filtered e-mail back into normal e-mail traffic is the only thing that it actually says it must do.

So let's go to the next slide. And this is where we see a portion of the specification that we are relying on for our construction. "The results of this analysis," which is the analysis of is it safe or not, "may be sent to the e-mail interface which reintegrates filtered e-mail back into normal e-mail traffic." So that's what it does if it is safe. "And which may send the model generator 240 each attachment to be analyzed further." It may add warnings. All those other things that counsel pointed out to you, it may do. If it is in a dependent claim, fine, it may also do that. But the e-mail interface, the only thing it actually does as set forth in the patent here, is that it reintegrates the filtered e-mail back into normal e-mail traffic.

So Columbia's construction is sort of just, you know, "We will find some helpful definition that's sort of hopelessly broad and could mean anything." But it is not really grounded in the specification. Our construction, we believe, is, based on this passage of the specification. That's really all we have on this.

THE COURT: All right.

MR. BIRNHOLZ: Very briefly. They pointed to the specification, Your Honor, and the one example of what the e-mail interface can do is quarantining. Quarantining is not reintegrating, so the claim should not be limited to that one element. Thank you, Your Honor.

THE COURT: All right. Let's move on to the next family of patents.

MR. SHEASBY: Your Honor, if I may, I have copies of the slides that I am going to show, if I can approach.

THE COURT: Sure.

MR. SHEASBY: Good morning, Your Honor.

THE COURT: Good morning.

MR. SHEASBY: The second family of patents that we are going to discuss, the '084, '306 patents, STEM from the work in the Columbia lab. And they are actually an interesting set of patents, because they have as their premise that some of the earlier work done at the lab was not as good as it could be. And I'm going to discuss the historical genesis of these patents because I think it really informs a lot of the claim construction disputes that we are going to hear today on these patents.

So we spoke about this tutorial that one of the standard prior art systems for analyzing viruses was to focus exclusively on malicious data. And there is actually two ways that you can focus exclusively on malicious data. One of those ways is the signature approach. And that's what the patentees are discussing in the background of their invention on this Slide 2. If a virus scanner's database does not contain a signature for a malicious program, the virus scanner is unable to detect or protect against that program. The prior art signature method is focusing exclusively on evil, exclusively on the bad.

There is another type of prior art system that counsel for Symantec pointed out, it is called, what they describe as a misuse system. I think that's the phrase they use. Now, a misuse system is also discussed in the background of the prior art. And what the inventors say about that system is it has the same failing, that what it ultimately collapses into is a focus exclusively on that which is evil. And if you haven't encountered the evil before, you are not going to be able to detect it if it appears anew. So we have this prior art construct, whether you use a signature, a fingerprint, whether you use some type of misuse system detection in which -- misuse system detection, the focus of this type of research was "Let's look at that which is evil and let's try to make sure it never appears again."

The patent talks about the fact that there is a group of researchers led by Professor Stolfo and others, actually, that spoke to the need to focus in a different direction. Not to blind yourself, to focus on what makes normal programs normal, what distinguishes normal programs from abnormal programs, and using that so that when you encounter something that you have never seen before, you are allowed, you are able to make a determination that this seems abnormal; this doesn't seem like how normal programs act. And because of that, because I'm suspicious of this, I'm going to flag it.

Creating a model of normal behavior to detect anomalies.

Now, what the inventors say, quite bluntly, is that these programs, these anomaly detectors, these models of normal behavior that have been developed, that have been developed by their lab, in fact, between the text on Slide 4 and Slide 5 they actually give the list of the publications that had developed these first generations of anomaly detectors, and many of those publications are actually by the lab itself. And what they say is, those fail in very important ways. And they list two in the specification.

The first one they list is computational overhead. The anomaly detection systems are so complex, they take so much resources to run, they run so slowly, that they are just too costly to be effective. And the second aspect of anomaly detection is that it is actually not as easy as you think to understand how normal programs act differently from abnormal programs. And the reason why it is actually quite difficult is because even normal programs act in a very irregular manner. It is tough to create a model of normalcy because there is not a recurrent pattern in computer system activity.

So what the inventors proposed, they proposed a number of strategies, but for our purposes today at claim construction, there are actually two strategies that I want to focus on. The first is that using a series of very elegant experiments, and by "elegant," I mean stripped down, very basic, they were able to determine something quite important. They were able to determine that focusing on a very, very particular location in a computer allowed you to effectively and efficiently distinguish that which is good from that which is evil. And that location is called the operating system registry. It is a unique structure in the program, unique structure in the operating system.

The second strategy that they focused on is something that involves probability. So what the inventors realized is that simply because you have never seen something before doesn't make it evil. It may just be new. And so what their system did is, when something that was not seen before appeared, they were able to assign a probability to it as to what the likelihood of it being evil was. So let me give you an example. If I see a program and it has a one in one-billionth chance of being malicious, well, that's a pretty low chance. So maybe I'm not going to shut down a computer system simply because I've seen something new before that has such a low chance of being malicious.

So now let's flip it over. What if I see something new that has a one in one-fourth chance of being malicious? That's actually a pretty high chance, and we know that malicious programs can do very serious damage.

We think of this as someone trying to get your bank account number. And for an individual, that's a very serious issue. But this goes beyond that. In other words, these systems are used to protect the Department of Defense, to protect our nation's secrets. This is serious as a heart attack for our nation as a whole. So a one in four chance is just too high to take so you will stop it. So these two strategies, focusing on these unique, unique structure in the computer system, creating this very elegant probabilistic model, allowed you to both create robust models, because the operating system registry was this perfect environment to be able to distinguish bad from good to what normal computer systems do, and it also allows you to make very intelligent decisions. Just because something is new, I'm not going to run away from it. I'm going to make a reasoned decision as to whether given it's new, do I need to be afraid of it. These are the insights that animate the '084 and and '306 patents.

One of the things that I think is neat about this family is, it is not -- you see that this is an academic lab that is doing the research. In other words, they see this failing, they see this problem, and they attack it directly. In other words, in the claims themselves, the solutions that they specify are there. "Gathering records of registry accesses." "Focusing on this unique structure." "Generating a probabilistic model." The insights that improve the old anomaly detection systems that they themselves created are in the claims.

So there are three terms that are at issue. We are going to do a ping-pong, so I'm just going to focus on the first term right now, "probabilistic model of normal computer system usage," and then "normal computer system usage" occurs subsequently in the claim, so Symantec, as is their right, would like construction of that as well.

There are two competing constructions of this term. And by competing instructions, it is really two ships passing in the night. So no party believes that terms that maybe have some jargon associated with them, what's a model, need additional construction. We feel that's something for right now that the experts are going to be able to inform the jury about. Columbia believes that the term ultimately and unfortunately that may need some additional construction is "probabilistic." Because during the meet and confer process it became apparent to us that there may be a dispute there, and we don't think there should be any kicking of the can down the road.

Symantec, in contrast, doesn't really want any additional construction of "probabilistic." What they want is to replace the word "normal" with "typical, attack-free." Columbia doesn't believe that claim construction should involve just replacing words with alleged synonyms, and we don't believe additional construction of "normal" is necessary. "Normal" is not a term laden with technical jargon. "Normal" is a term that has a meaning to folks in their everyday lives. It is not used in an idiosyncratic way in the patent. It is not to suggest that the challenge of determining whether a given system is normal or not is going to be easy or it is going to be something that a jury is going to be able to do without an expert explaining the system to them, but it is not really about claim construction. It is about something else. It is about comparing the system and the nature of that system to the understanding of the term.

So let's jump right into the three disagreements. The disagreement on "probabilistic." Symantec's original position was that "probabilistic" meant something called "based on a probability density function." The problem with that is it once again kicks the can down the road. I actually don't know what is meant by a "probability density function." And it is just going to create another debate the morning of expert reports. And we pointed out to Symantec in the meet and confer process, we really didn't think that was right. We proposed what we thought was a correct construction of "probabilistic" and Symantec responded to us and said, "Okay, let's just say it is the plain meaning and move on." We said, "That's okay, we have no problem with that. But are you saying, do you think the plain meaning is a probabilistic density function?" They say, "We reserve the right to say the plain meaning is a probability density function." I think that's a recipe for dispute. I do think we need to engage it if that's in fact where the dispute is going to lie. "Probabilistic" as used in the intrinsic record is consistently described as a model that provides probability. I don't think that is subject to great debate. "Probability" also is not a term that has an idiosyncratic meaning in the specification. This is a standard accepted definition of "probability" on Slide 13. And if you look in the specification you see the resonance to that standard accepted definition. You see the consistent repetition of the probabilistic model creating a likelihood. So Columbia believes that its construction shows fidelity to the specification. And the probability density function proposal, our concern is that that just delays to a later point in time some fight about what that language is going to mean.

So I want to jump to the second area of dispute. I'm now on Slide 17, Your Honor. This relates to redrafting the claim by replacing "normal" with "typical, attack-free."

So there are portions in the specification that uses the words "typical" or uses the words "attack-free." What's significant is that we know the inventors knew how to write the words "typical, attack-free" because they do it in discussing certain embodiments in the specification. What they did not do is use the phrase "typical, attack-free" in the claims. In the claims, they make reference to a "model of normal behavior." Not a "model of typical, attack-free behavior."

And this is another symptom of, I think, kicking the can down the road. Replacing "normal" with "typical, attack-free" doesn't do anything. It doesn't add clarity. It doesn't resolve any debate. It just leads to a further fight about what "typical, attack-free" means, in particular, what does "typical" mean, and in what context, as to whom. We are not really engaging in anything substantively; we are just replacing words. The Federal Circuit is actually pretty conscious of this. If you read the C.R. BARD decision, this is one of many, one of the things the Federal Circuit is not shy about reminding courts of is that claim construction is not claim redrafting. The process of replacing one word with an alleged synonym doesn't advance the claim construction process. All it does is, it fails to show fidelity to the primary source of claim construction, which is the claim itself. If we need to have a discussion about what "normal" means, let's have a discussion. Let's not have a discussion about what the words "typical, attack-free" mean because those aren't in the claims.

The Federal Circuit also points out this challenge in the PPG decision. Claim construction is not the process of setting out the test that's going to be used to assess infringement. Some words only have a certain amount of precision associated with them, and the Federal Circuit recognizes that. The step of taking that to the infringement analysis is for the jury, not the judge. And this is a standard that I don't think is always applicable, but I think it may be applicable here. In other words, "normal" is used consistently in the specification. There is not a test that says this is the definitive test for normal in the specification. It has its plain and ordinary meaning. And the task of applying that plain and ordinary meaning is really for the jury. It is not a question of claim construction.

So now let's go to the third dispute, which is Symantec's request for negative limitation excluding any consideration of abnormal access information. Let me unpack this. So what Symantec appears to be saying is, and I think the reason why they want the "typical, attack-free" language to be put in the claim, is they are going to subsequently say, "And typical, attack-free, if your model is a model of what is typical and attack-free, you will never, ever, ever consider any other data in constructing that model other than data on normal activity. You will blind yourself to everything else.

When we spoke earlier this morning I spoke about the fact that creating a model of what it means to be an apple, how much more robust and meaningful that model can be when you think about not just what an apple is but how an apple is different from an orange. It creates a more meaningful model. And what Symantec is saying is that the inventors somehow told the workers who would read this patent, "We want you to blind yourself on every other type of data. You can consider data based on normal activity, but do not consider any supplemental data on abnormal accesses when constructing the model." The standard for importing this type of negative limitation, the claim involves manifest exclusion, a clear disavowal, a clear and unmistakable disclaimer. None of this appears in the patent. And this is not a "Gotcha." This is not saying "Ha Ha, we didn't disclaim it all the way." The reason why none of this appears in the patent is this patent is not about whether you consider supplemental abnormal information. It has nothing to do with this patent. This patent is about very important scientific observations about the nature of the operating system registry and about the powers of probabilistic models. And those important insights can be applied whether you use 100 percent normal data or whether you supplement your normal data with abnormal data.

So let me tell you what I mean. Let's assume that you have a training set data with three programs. All those programs are normal. Function A, Function A, Function A are all performed by those programs. Think of the Function A as a feature of what the program is doing. You can say, "I'm going to create a model of normal behavior and that model of normal behavior is that if you perform Function A you are normal." What if you have a Program 4, and that Program 4 is abnormal? It performs Function A, but it also performs Function 1. Having that information about how supplemental abnormal programs work allows you to create such a more meaningful model of normal operation and your model of normal operation is that Function A is normal as long as it is not in combination with Function 1.

So that's really the dispute. That's, I think, the bidding. Symantec is saying the inventors clearly and unequivocally told workers to blind themselves totally and completely to anything other than this. Even though it is absolutely undisputed that the prior art made clear, the publications of these own inventors, that there is meaning and power to considering supplemental abnormal data. And even though this patent has nothing whatsoever to do with excluding supplemental abnormal data.

So I think what Symantec is doing is really confusing two questions. The first question, and it is an important question: What is the subject of your model? The second question: What what is the information you use to construct that model? They are distinct. Question 1, what is the subject of the model in the patents? The claims answer this. A model of normal computer system usage. Question 2: What information is mined to construct the model on normal computer system usage? Well, the patent is also explicit about that. If you are constructing a model of normal system usage, you, of course, have to consider the normal operations of programs. That's of essence. And the specification makes that clear, that you need to do that. And in particular, you need to determine a very, very particular type of normal activity. The specification says, "I want you to consider normal activity that relates to accesses to the registry," that unique feature that is animating the research that led to this patent.

But what the claims don't do, the claims don't say, "And I want you to exclude all other information." Claims list what must be present. Claims don't list the -- the fact that they list something that must be present doesn't mean that other things are excluded.

So in Symantec's responsive brief, I think they did something interesting. What Symantec said is, "Well, we are not arguing there is a disclaimer. We are not arguing that there is any statement that says you can't consider supplemental abnormal data in constructing your model. Our problem is there is no embodiment that exists in the specification that uses supplemental abnormal data."

So the Federal Circuit has also spoken clearly on that issue. What the Federal Circuit has said, en banc, in PHILLIPS, is if you assume that every single embodiment in the specification, every single one said openly and clearly, "I want you to exclude all supplemental abnormal data, when you are creating your model of apples, you look just at apples and you blind yourself to oranges," if every embodiment in the specification said that, what the Federal Circuit en banc in PHILLIPS has said is, that is not a basis for importing that limitation into the claims.

Once again, this is not a gotcha. This reflects how patent attorneys write patents. The purpose of a patent is not to lay out all the different ways in which these new inventions can be applied to the prior art. It is not to list all the different anomaly systems that can take advantage of the probabilistic models, that can take advantage of tracking registry system accesses. It is to give to the public, give to the world, that which is novel, that which is important and is new. That's reflected in the claims.

Registry system access. Probabilistic model. The inventors of this patent did not give to the public the idea of blinding yourself to supplemental abnormal information. That's not their invention, and it is not what they are teaching.

So Symantec makes another interesting argument. They say the ordinary meaning of the term "model of normal behavior" is the exclusion of any supplemental abnormal information. So I think what's interesting about this is what Symantec has ended up doing is they have ended up conflating those two questions that I spoke about earlier today. Symantec's proposal for the construction of model of normal behavior is a model of typical, attack-free behavior. Even their discussion, their construction, is defining what the model is. Their construction doesn't define what the model includes, what is used to create the model. And so for them to say, "Well, our argument is about the plain and ordinary meaning of a model of normal behavior" doesn't really answer the question, because that's about what the model is. It is not about what is used to construct the model.

So there are, I think, three points that I think address this argument about the normal, ordinary meaning of a model of normal behavior. The first is the inventions do not exclude supplemental abnormal information. We will talk about why that's the case. The second thing we will talk about is the prosecution history makes absolutely clear that systems that use supplemental abnormal information, in addition to normal information, to construct their models, are covered by the claims. And the specification actually teaches that supplemental abnormal information can be used.

So let's jump right in. I showed this slide previously, but I think it is really important to emphasize. The drawbacks, the failings in the prior art that are described in the specification have nothing to do with the use or absence of supplemental abnormal information. Nor for that matter do the insights that are described in the specification.

When the specification talks about prior art, whether it is the signature method or whether it is the intrusion method, the failing of those methods is because they blinded themselves to normal information. They focused exclusively on abnormal information. That was the failure. That was what they thought was deficient. The inventors never said, "Oh, we think it is horrible to supplement your normal information with abnormal information in every single situation." There are situations in which it may be appropriate not to use supplemental abnormal information. There are certain stripped-down algorithm designs in which your algorithm can't actually handle supplemental abnormal information. It is not complex enough. But to say there are designs in which you would not supplement has nothing to do with a clear and unmistakable disclaimer of that strategy in every single situation which the Federal Circuit specifies.

So one of the things that Symantec does and they do it in each phase of their argument for the families is that the original provisional applications for these specifications, they weren't fancy, written provisional applications. They were in most situations just manuscripts that the inventors had written, that they were excited about, that they wanted to file a patent on, so they wanted to get it filed. This was very important because in Europe, the standards for anticipation are much different than in the United States. And in many, many situations, as soon as the work was done, Professors Stolfo and Keromytis were on a plane to present the work at a conference. So that would be a problem in Europe, so they filed these manuscripts.

What I think is neat about these manuscripts is they weren't written by patent attorneys, they weren't written by people looking to define terms or do the stuff you need to do to make a good patent application. They were just written with the science in them. And if you look at the conclusion of this provisional application, the conclusion of this provisional application, this article, what they were excited about was the power of the registry access. In fact, they were so excited about it, they said, "We think you can do this, this anomaly detection, looking only at registry access information." What they did not say in the conclusion is to blind yourself to supplemental abnormal information. Because that's not their invention. That's not the intent of this patent.

In fact, they were blunt. They said, "We used very unsophisticated algorithms in performing these experiments." And they did. But what they also said is, "We expect that you will do a much fancier job of this than we did. Because our goal was to capture the insight, not to use all the different embodiments -- not to disclose all the different embodiments that could practice this insight."

Let's talk about the prosecution history now. So during the prosecution of the patent, the Patent Office rejected our claims over a reference called Chong. And if you read Symantec's brief, they say, "Well, Chong is an interesting reference because Chong actually uses mixed data. It uses normal data, it supplements it with abnormal data, you use all the data to construct a model." And they said the fact that the Patent Office allowed our claims over Chong shows that our claims don't cover the mixed data situation. But there is something very interesting, and it relates to what actually happened. The PTO allowed our claims over Chong not based on this issue of supplemental abnormal data. To the contrary, the PTO felt that the fact that Chong used supplemental abnormal data that he knows that it was within the scope of the claims, which would only be possible if those claims covered the use of supplemental abnormal data. The PTO allowed the patent over Chong because Chong didn't disclose tracking registry accesses. That was the basis for the allowance.

The insight, what made them so excited when they wrote that conclusion in that manuscript, was the basis for the allowance. Chong was not anticipating because it didn't track registry system access. There was nothing, no statement that the claims didn't cover Chong because Chong used supplemental abnormal data by the Patent Office. In fact, the Patent Office concluded, we were out of luck on that limitation.

So let's dive in and get a little more information. So when Columbia received a rejection based on Chong, its immediate reaction is that, "Well, Chong is really only considering attack data. If you read the specification, it is really focused on just considering attack data. And that is different from the claims. The claims require you to consider normal data." Absolutely. But the PTO said, "You know what? We disagree with that. We think Chong covers both abnormal and normal data together." And Symantec said this as clear as anyone. They said, "The dataset in Chong used to generate the models includes data representing both typical network behavior and attacks." That's Symantec's position. Chong was rejected, the claims were rejected as anticipated, which means each limitation must have been in the Chong reference over Chong, even though this is what Chong disclosed, mixed data.

The PTO's conclusion was impossible if the claims didn't cover normal and abnormal data. What the PTO said right there, that's the basis for allowance. Not mixed data being outside the scope of the claims.

Registry access. So let's jump into the third section of our discussion, which is what the specification says. So there are embodiments in the specification that clearly use abnormal information to build out the model. There are incorporated articles in the specification which discuss the construction of anomaly detectors, models of normal behavior with mixed data. And of course the incorporated '342 application uses abnormal information as well. I want to focus in the first instance on the first bullet point, because I think the last two are discussed at length in the specification -- in the briefing. I want to give two examples.

So one of the passages that Symantec quotes from extensively is this construction of an embodiment in which in the first steps they are only using clean data, 100 percent clean data, to build out the model. And Symantec quotes from that saying, "See, Aha, they are only using clean data. That's what the claims are limited to." Then Symantec quotes this sentence at the top of Slide 46, "Anomaly detectors do not operate by looking for malicious activity directly." And they are right. That's not how anomaly detectors models of normal behavior work. They try to understand how normal divergence from malicious in order to make a determination.

But then the inventors go on to discuss a problem with the model they are constructing, this model that only used clean data in the first instance. What they say is, "Well, one thing we have noticed is that we are tracking the registry, and when new programs are added to the registry, when new programs are added on to a computer, the registry activity is intense, and that is going to appear as an anomaly, because it really doesn't occur very often with 100 percent clean data." And we have a problem with that. And the problem we have with that is that there are many instances in which perfectly benign activity will add new programs to the computer. So when you plug in a Sony phone into your computer for the first time, you will get what's called an Install Wizard saying, "I'm going to add a driver to the program." So that is a graphical representation of something that's occurring in the registry. An alarm has gone off in the registry, saying, "I'm adding a new program."

So what the inventors pointed out is that if they just stopped there, left the model just focused on 100 percent clean data, they would always be triggering as malicious the addition of new drivers to the program, which means no one could ever use the computer, because every time I plug in my phone, a new phone or a new camera, the computer would shut down as detecting a malicious attack.

So what they say is, "Here is what we have noticed. We are going to look at how malicious programs act differently from normal programs. Malicious programs often install quietly so that the user does not know the program is being installed." So they have seen the problem, they have seen the failure with their model, a model that focuses only on 100 percent normal data, and they said, "We are going to adjust the model to take into account the differences between normal and abnormal. And the difference that we have seen is malicious programs install quietly, so we are going to create a new rule, and that new rule is that in our exemplary embodiment, the algorithm is programmed to ignore alarms while the install program is running." If the install program is running it is not a quiet installation. Everyone knows the program is being added. And that's not dangerous. The ones you are scared of are the ones that are secret.

This is a very stripped down example, because remember, the inventors were doing this purely for the purpose of creating a design that they could test a hypothesis with. But even in this stripped down example, they are not blinding themselves to abnormal data. They didn't write the specification and say, "Well, we constructed a model of normal behavior and it is really not very good because it catches every time you plug in your new camera to a computer. Oh, well." They said, "No, we are going to enrich this model, we are going to enrich this model with information on malicious behavior." This is one example. It is one example. It is not a massive example. This is not a massive experiment. It is a very, very small experiment.

But this shows something. You don't blind yourself. They looked at apples, then they looked at oranges, and they said, "How are oranges different from apples?" And they used that to make the model more robust.

Let me give you another example. The design in the specification, very stripped down design, set thresholds, probability thresholds. If you are above the threshold, the risk of you being malicious is so great we are going to cut you off. If you are below the threshold, we are going to let you through. So once again, they are originally constructing the model using clean data, and you will see a lot of references to that in Symantec's presentation, is my expectation. And that's right, they usually start with clean data in their stripped down embodiment. But what they do is, they then test the model to adjust it. They expose the model to normal and attack data, data they know is normal and data they know is an attack. And they run the analysis. You see those numbers on Slide 50? Those are really setting thresholds, probability thresholds. And what they are experimenting with is they are experimenting with is, "Where should we set the threshold on the model." I'm on Slide 51. You can see this expressly. They are texting mixed data and using that information that they gather from the mixed data to refine the model, to adjust the thresholds. It is once again a very minor example, but it is an example in which you don't blind yourself to data.

So the reason why these are such minor examples is not hard to gather. The purpose of these experiments, the purpose of this patent, is not about whether you use mixed data or 100 percent clean data all the time, whether you supplement, whether you adjust, whether you blind yourself. That's not what these experiments are about. These experiments are about defining core principles of the power of probability in this system, about the power of the registry system accesses.

So two other points: Incorporated articles use abnormal information. I won't go into this in great detail because it is extensively set out in the briefing unless Your Honor would like me to address it in greater detail. I think it is important to point out what the inventors did is they said you can build anomaly detectors based on models of normal behavior, and we are going to give you a ton of ways you do that. They cited their own articles. Many of those articles discuss the efficacy of using mixed data. That's a completely different area of research, and the inventors are actually one of the forebears in presenting the power of mixed data. But they are not going to touch it because it is not what this patent is about. This patent is about something different: Whether you used mixed registry access data, 100 percent clean registry access data, the key for the inventors is you have to look at the registry. The same holds for the '342 application, once again, I won't repeat what's in there, Your Honor, unless you would like me to address it in any particular detail.

So I think with that, what I would like to do at this point is save a very short period of time in rebuttal and pass the podium off to my colleague.

THE COURT: All right. We will take ten minutes.

(Recess taken from 11:31 a.m. to 11:46 a.m.)

THE COURT: All right.

MR. NELSON: Thank you, Your Honor. Dave Nelson on behalf of Symantec again. So let me get right to this first construction issue. We can go to Slide 23. In the book we gave you all the slides together so everything should be there that we are referencing for the most part today.

So let me address this, the first issue, what I think is the more critical by a long shot issue with this construction, and that's the definition of "a model of normal computer system usage." Now, what Columbia is saying is that doesn't need to be defined. "A model of normal computer system usage" requires no construction, because everybody knows what it is. It is just normal and everybody knows what normal is. That's a problem. We've got a big problem in the briefing and the argument you saw here leads you right to what that problem is, Your Honor. Because the goal is not in claim construction, well, if I use words that somebody in context -- I mean out of context might know what it is in some certain context, then there is no need to define it. That's not what the exercise of claim construction is. What we are supposed to to be doing is looking at the terms that are used in the context of the description of the invention that's provided by the specification.

So what we have here, and you can see from the argument and from the briefing, we have what I would call, because patent lawyers always abbreviate cases, it is an 02 MICRO issue. O2 MICRO is the Federal Circuit case that says when there's an issue of claim scope, it is up to the

Court to resolve that issue. And I think we have a very fundamental issue of claim scope here. Because with respect to the Symantec construction and what we believe a model of normal computer system usage is, it is one that doesn't include attack data. It is based on normal computer system usage, which is attack-free, typical, attack-free data.

What do you hear from Columbia? You don't hear anything because they don't want it to be construed. But they say, "Well, it can include attack-free data, or excuse me, attack data. It can include anything as long as I can have an expert sit up there and say, 'Well, it is normal, it has some normal data in there,'" and that is a claim construction issue, Your Honor.

So let's go to the next slide, 24. Columbia, they are trying to turn around what's really going on here and trying to say that we, Symantec, are arguing for a disavowal. In other words, that they disavowed the use of attack-free data, or excuse me, attack data when creating a model of normal computer system usage. That's not what is going on here at all. What we are trying to do is in the context of the intrinsic record look at how the inventors themselves described and defined what a model of normal computer system usage is. That's the issue here. There is no disavowal being argued. So that is an attempt to turn around what's going on here, to try to support their position that, Your Honor, I think the term used by my colleague was "kick the can down the road." That's exactly what they are asking you to do. "Kick this can down had road so I can have an expert sit up on the stand and say, 'Hey, I know normal when I see it, and this is it.'" So this is not a disavowal situation at all.

So now let's talk about, if we go to the next slide here, what it is we are doing. Because as I said, Your Honor, and you see we have a few cases, they are highlighted in the brief, but what we are trying to do in the context of claim construction is to look at how the inventors described their invention. Look what it is, look how they use these terms in the context of the specification, and give those terms a meaning in that context. Not just divorced from the specification, not in the air so somebody can come along and argue later, "Well, this is normal." That's what we are trying to do here, Your Honor.

So let's go now and start looking at some of the ways that the specification talks about the invention, which is an anomaly detection algorithm. I don't think there is any dispute there. That's what both of us said in the background argument. So here, this is from Column 2, Lines 34 to 37 of the patent, and this is, our citations, Your Honor, are to the '084 patent in these slides. The specifications are pretty much identical, and therefore, we didn't cite to both of the -- the '306 is a continuation, so it is primarily where the differences are maybe in some of the claim language. So that's why the citations are that. But here we see, "Anomaly detection algorithms may build models of normal behavior in order to detect behavior that deviates from normal behavior." That's what they are trying to do. That's the way the inventors set up their particular system and distinguish if from some of these other signature-based systems, misuse systems you heard talked about earlier today.

So here then, if we go to Column 7, "Anomaly detectors, such as Anomaly Detector 16," if you look in the patent, that is just a block diagram of the description of the invention here, "do not operate by looking for malicious activity directly. Rather, they look for deviations from normal activity." So that's the way these systems work. Let's figure out what's normal, and then let's look for deviations from that normal activity to determine if something is an anomaly, in other words, in this instance, an attack.

So the patents also talk about, if we go to the next slide, how you do that. This is from Column 6, Lines 26 to 32. And it says: "If a model of the normal registry behavior is trained over clean data, then these kinds of registry operations will not appear in the model, and can be detected when they occur." So in other words, I look at the clean data, I look at the data to create my model of this is the normal behavior for this system. If I see something different from that, then I detect that as an anomaly. That's the way these systems work.

So let's go to Slide 29. And here, in the example that's provided, this is in Column 15, 4 through 16, about how that model was created, it is specifically stated to be generated by clean, meaning attack-free dataset. This is approximately 500,000 records that were used here. So that's the example that's given in the specification.

Now, you heard counsel at the end of his argument say, "Well, there's some other systems that talk about using abnormal data in order to create the model." No, no, that's not right. If you go in and look at the patent, and you'll see this from the briefing, when they talk about creating the model, it is clean, attack-free data. What counsel was talking about was the second step when you are doing the comparison of some observed activity, you can adjust the threshold. That's determining how unlikely the event is going to be. Because remember, it is not just a simple yes, no. There is this probability that's associated with it. And so the things that counsel was citing to were directed towards that, the adjustment of this threshold when you are doing the comparison to see what kind of behavior you are going to tolerate before you trigger an alarm that there is an attack, not to creating the model that's used to establish the baseline to which the observed activity is compared.

So now, let's go to Slide 30. And this may require a little bit of background. There was some talk of this during counsel's argument, but let me provide a little background here. There may be a question in your mind, Your Honor, why the parties are talking about a '342 application, why the parties are talking about perhaps an earlier provisional application. Let me give you a little bit of background. Both this '342 application, it was an application that was filed the same day as the application that eventually gave rise to the '084, and then as a continuation, the -- I'm forgetting the number -- '306. Six numbers is too many to keep in my head for me, Your Honor, I'm sorry. The '306. So it was filed the same day. It is also incorporated by reference, and you saw an earlier citation in the argument of how that was done. So this description was incorporated into reference, the '342, so it becomes part of the intrinsic record. It is part of the things you want to look at when you are trying to figure out what the inventors told those of ordinary skill in the art about the invention and how they described some of these terms.

The provisional application, the same. So, you know, sometimes, Your Honor, with a provisional application, you may file it, and then within that year period, you go ahead and file the full utility application with the claims and everything, and you can just cite back to that provisional application and say, "Well, I'm claiming priority back to that date." Then there may be a dispute about exactly what is supported and what's disclosed. That's one way you can do it.

And there was some cases cited by Columbia saying, "Well, don't look to that provisional because there could be lots of changes." Well, those are cases in that, when we read those cases, in that context where there may be a number of changes to the final document, and there's discrepancies between the way the final patent specification discussed the invention or described certain terms, and the way it is done in the provisional. This is a different situation, because we actually have the provisional as well as this '342 application incorporated by reference. So in other words, they are telling the public, "Go back, if you need more description of these things, and look at these applications because they will provide you additional description about some of the things that we are using."

So in its brief, this is in the response brief, Columbia looks, they cite a number of things from this '342 application, the one that's filed the same day. And one of the things they do is to say, "Well, of course, Symantec must be wrong, because that '342 application describes a data warehouse." In other words, where all these records are maintained. And if you look at the face of the '084 patent, for example, you will see, Your Honor, for example, Box 18 that's labeled a "data warehouse." There is no dispute there. That is where information is taken from in order to create these models, in this case, the model of normal computer system usage, so that you can later do the comparison of the observed activity.

Well, what Columbia says is, "In that application," the '342 application in this context, "that data warehouse is described to have both clean data and attack data." No dispute there. It is. But that same '342 application is also very clear, and if we go to the next slide, 31, this is Paragraph 69, what it is teaching is that in order to build certain models, and we see right here the paragraph I have highlighted, different types of model building algorithms require different types of data. And I'll go into this a little bit more, but the '342 application, it says explicitly, "Okay, a data warehouse, I have a whole bunch of records in there. But the records that I use in order to create a model have to be tuned, selected, filtered," what term you want to use, "to the particular algorithm that I am going to use in order to do my detection."

So the fact that the '342 application may talk about a data warehouse that has other records in it is really irrelevant to the question. And in fact, the '342 application is very clear that the particular type, the anomaly detection algorithm that we are talking about here in the '084 patent requires clean, attack-free data. And let me explain that a little bit more.

So here, Your Honor, if we go to Slide 32, this is also from that same '342 application incorporated by reference. It says that "Anomaly detection algorithms train over normal data to create a model of normal activity. These algorithms need to train over data that contains no intrusions." "No intrusions." In other words, no attacks.

So it says that explicitly. Now, does that same '342 application describe other systems? Sure. It describes things like unsupervised anomaly detection systems. Another one in there that "may use clean and dirty data." But that's a different system from the one claimed in the '084 patent. And I will show you that Columbia agrees with that in a moment, Your Honor. So that's a different system. They talk about misuse detection systems. That was one that both myself and Columbia's counsel described as being something that the inventors talked about in the prior art. I think during Columbia's argument they said one of the criticisms of these misuse detection systems was that they were computationally intensive, I think was the citation they had. And that that was a problem.

Well, if we look at Paragraph 102, and you actually don't have this slide, we will get this to you, Your Honor, but this is Paragraph 102. This is describing one of those misuse detection algorithms. Not what's claimed in the '084 patent, but a different one that everybody agrees is prior art. What does it say? It says, "Misuse detection algorithms train over normal and attack data. Using this data, these algorithms build a model that can discriminate between attack records and normal records."

But if you look down, it talks about those disadvantages, the same disadvantages that counsel pointed to. This data is very expensive to obtain; it may not be portable; it requires labeling; training in data. In other words, computationally intensive. So yes, there are misuse detection systems that are described here that use both norm normal and attack data, but that's not the system that's claimed. In fact, that's one of the systems that the inventors criticized in the prior art because it required this computationally-intensive approach in order to obtain. Yet, Columbia would tell you that we could do exactly those things with the model of normal computer system usage, it contained normal and attack data. But that's what they criticize the prior art having. So what we want to focus on is the anomaly detection system, because that's what's claimed here in the '084 patent.

If we go to the next slide, 33, back in the slides that you have, Your Honor, you will see that Columbia agrees with this point. This is from Page 20 of their response brief. And this is in reference to that same '342 application that we were discussing, Your Honor. It says: "The '342 application confirms that its diverse datasets can support a wide variety of different intrusion detection systems." I talked about that. Sure, the '342 application talks about that. But let's focus on the specific one that's being claimed here in the '084 patent. One example is the Registry Anomaly Detection, RAD, system, which is described in greater detail in the '084 and '306 patents. That's Columbia itself, the RAD system, that's the one to focus on. It is the anomaly detection system. If you look actually in Column 4 of the '084 patent towards the bottom right when we begin the detailed description of exemplary embodiments, it makes the same reference to that RAD, Registry Anomaly Detection system as well, just as further confirmation.

So I pointed to you a minute ago, Your Honor, and that was in Slide 32, that this is from the '342 application, that "Anomaly detection algorithms," according to this '342 application of which this registry anomaly detection system is one, "train over normal data to create a model of normal activity. These algorithms need to train over data that contains no intrusions." So the application, the '342 application, is very specific that the system we are talking about, the anomaly detection system, this one specific to the registry, trains over data that contains no intrusion. That's how the model is created.

Now, I also talked about this provisional application which is incorporated by reference. That's the '857 application. If we go to Slide 34, Your Honor. Now here, this talks specifically, uses the RAD term, right, and it says, "RAD generates a model of normal registry activity." And then further down in the application, it says, "In order to evaluate the RAD system, we gathered data by running a registry sensor on a host machine. We used only attack-free data for training." So very explicit. In the provisional application that's incorporated by reference, that this RAD system, which Columbia agrees is the one that's described in the '084 patent, is one where attack-free data was used for training.

Now that, Your Honor, is the reason, that, what I have just gone through, the reason why you do have a serious claim construction dispute here and one that the Court needs to resolve. Because there is no dispute between the parties at all that you can have different systems, different types of detection systems that may use different types of data to create a model. They might use dirty data -- or excuse me, they might use attack data, they might use clean data, those kind of things. But the one we are talking about here, the one that creates a model of normal computer system usage, the Registry Anomaly Detection system, is one that's trained on attack-free data. That's a claim construction issue. Columbia disputes that. They want to be able to argue down the road, "No, it can include attack data as well." That's why we have the claim construction issue, Your Honor.

Let me go to Slide 35. Counsel mentioned, and this is covered in the briefs, but counsel mentioned that there were also additional papers, this particular paper is one that Dr. Eskin is one of the named inventors on the '084 and the '306 patents. So this article is also incorporated by reference. Many things were incorporated by reference. This article was. And so what Columbia has argued in its briefs is, "Well, this paper recognizes that we can create certain anomaly detection systems that train, in other words, create their model, using both attack data and clean data." Right? The problem and the reason why we cited this paper is, we are not arguing that nobody could have done that. That's not the point. We are trying to figure out what was claimed here in this particular patent, the '084 and the '306 patent, not what somebody could have done or what they had the capability of doing outside the four corners of this patent. That's not relevant to the discussion.

But what this paper does do is show very clearly that the inventors knew the difference between clean data and noisy data, in other words, data that included attack, and on the other hand, the attack-free, the clean data. So, and why does the paper explain, so if we go to the next slide, the paper explains why it is important to use the clean data in these anomaly detection systems because it says, "If there is an intrusion hidden in the training data, the anomaly detection method will assume that it is normal and not detect subsequent occurrences." So in other words, you have to know, to train the system to create the baseline, that you really are doing what's normal. If you include a whole bunch of attack data, then, according to this paper, that's going to be viewed as normal activity so that when you go out there again, now you are running the system out in the field, and what you are trying to do is prevent attacks. Well, you are going to see another attack but since that was in your training data to create your model, you are going to think that's normal. According to the paper, that's no good because now you are not going to detect that, you are going to be attacked.

So go to the issue that I just talked about. It is Paragraph 37. So here, it talks about "Traditional anomaly detection techniques focus on detecting anomalies in new data after training on normal (or clean) data." So right here in the paper it is incorporated by reference. Normal data as understood by these inventors is used consistently in the specification, used consistently in the applications that are incorporated by reference, and is used here in this paper that's incorporated by reference, to mean clean data. Right.

When they go down further and in this paper, we don't dispute that, they present a different method. "We present a mixture model." So in other words, explaining the presence of anomaly in the data. It is not normal. They describe it as a mixture model. Earlier in the title they talk about it being noisy data when it includes attack and clean data. So normal, according to all of these references that we looked at, means clean. Our definition, attack-free. We think that's -- and you saw plenty of citations that I gave you where it is described to be attack-free. But that's what "clean" means in this context.

That's the terminology that the inventors chose to use in their claim, right? They could have said, you know, "creating a mixture model," or "creating a model of computer system usage using noisy data" had they described those things. They didn't. They used "model of normal computer system usage." And according to all these references, that means exactly what we say it means, which is free from attacks, Your Honor.

Now, go to Slide 39. Here is another issue. This is why I do think it is a claim construction issue. But from the argument you heard from counsel and as well as what we see here in the briefs, what Columbia is saying, and this came across, I think, particularly starkly when they were arguing about the prior art in the prosecution history and how that was distinguished and what the basis of that distinguishing was of the prior art, and what Columbia says in their brief and what I heard here in the argument today is, "Well, see, that was distinguished because it only included abnormal or attack data. Our model, it is not solely based on that. It also has normal data in it."

Well, think about that, Your Honor. So how do we decide what's an acceptable amount? So if I have a model that has one record that's normal, attack-free, and 99 -- I use percentages, it is easier -- one percent normal and 99 percent attack data, well, according to the argument you heard they distinguished the prior art, that would be a normal computer usage system model because it is not completely based on abnormal records. Two percent, three percent, four percent? How do you define how much normal data needs to be there, under their argument about what normal computer system usage is, to decide what normal computer system usage is? It is completely boundless.

There is no way that one, and under the NAUTILUS standard, that's the Supreme Court case that came down on indefiniteness, both the claim and the claim construction have to tell those of ordinary skill in the art not only what's inside, but what's outside, where the bounds are. They have to reasonably tell you that. With this construction or the argument that Columbia is making for you, you don't know what it is. You know it is somewhere between one record and 100 percent of records. But where that line is drawn, who knows? And the HALLIBURTON case we cited here bears a little bit of discussion, Your Honor. That's 514 F.3d 1244.

But the issue there was one where claim construction was proposed, and they wanted to have the construction of fragile gel, what fragile gel was. And the construction that was offered didn't provide any guidelines of when a gel became fragile, you know, what was the requisite degree of fragileness, is what they said. That's the same issue we have here with the argument that Columbia is making. The line is very bright under Symantec's construction. It is consistent with the specification, it is consistent with the way the inventors have chosen to use that term in the context of their invention, and also consistent with the way the inventors have distinguished other systems than the ones that are claimed here. Columbia, on the other hand, we don't know. We don't know where that line is. And that's a further problem, Your Honor. Now, the last -- and I have a citation on Slide 40 that just highlights that again, but I think it is apparent from the argument. But the point being that it can include additional data. But how much additional data, we don't know.

Now, probabilistic. That was the last, if I go to Slide 42, Your Honor, that was the last piece of this. I don't think we have a major dispute with respect to this, the probabilities, the likelihood that the event will occur or condition will be present. You know, probability. I think the only -- we have offered the construction if you look at the whole term which would be based on a probability, which is what this invention describes. Columbia offers this construction of "employs probability." Here is the problem that I have with that, Your Honor. And you may say that this is patent lawyers, you know, how many angels can dance on the heads of a pin or whatever the analogy is. Those sound similar to me. But here is the problem I have with what I think we may hear down the road based on some of the infringement contentions we have seen in this case.

"Employs probability" conveys the idea that as long as probability is used somewhere in the determination, then it employs probability. But that's not what this is. This invention is you create your baseline model of normal computer usage, you observe the activity, compare it to that baseline, and then based on that probability, on a probability determination, the comparison of how likely that event is to be normal, you make your determination as to whether something is an attack. So the way I would look at this, Your Honor, so let's say you have a baseball manager out there, major leagues, and you know how baseball is a game of statistics and everybody is always looking at how does this guy do against lefties and all of those kind of things. So he has a situation, the manager has a situation where he looks down the bench to see who he has got because he is not going to let the guy who is supposed to go up there to bat bat. He is putting somebody new in a game situation. He looks, "Okay, what's this guy's average," he asks the coach, "What's this guy's average against this pitcher?" In other words, the probability that he will get a hit. "Well, he is, you know, one out of three. 333." "How about this other guy?" "He is 250." So he is less. Well now, the manager, if he was making that decision based on probability, he would say, "Okay, this guy, he has the better chance, given history, he has the better chance to get a hit. I'm going to put that guy in, the 333 hitter." But that's not what the manager does. The manager is like, "Oh, it is not that big a difference and I just have the feeling that this 250 hitter is going to have a good day, the gut feeling." He says, "I'm going to put that guy in." He goes with that guy. Well, that's not a decision that's based on probability. In fact, it is contrary to what the probability evidence would say. But it certainly employed probability in the decision process because he wanted to see how these two things are.

And that's the problem that I have with their construction, Your Honor. I think it is too amorphous. I think it invites the idea that this decision doesn't have to be determined based on a probability comparison. It just has to be somewhere in the process. And so that's the reason and that's the basic dispute, I believe, that we have on that term, Your Honor.

So I'll turn it over to my colleague. I think that's all I have to say on this, and I thank you for your attention, Your Honor.

THE COURT: All right.

MR. SHEASBY: So I think when we listen to argument, sometimes it is most important to remember what's not addressed in argument as opposed to what is addressed. And I think that -- give me one second, Your Honor -- I think the first thing that you didn't hear is, you didn't hear any discussion whatsoever about the fact that during the prosecution of this patent, the reference -- the claims were rejected over Chong, Chong disclosed the missed system -- mixed system, excuse me, that mixed system was found to be anticipated and the claims were allowed not based on the presence or absence of a mixed system, they were allowed based on a registry access. You heard no argument whatsoever from counsel for Symantec addressing that point.

He brought up Chong, I give you that. He brought up Chong as part of a new indefiniteness argument which you will find nowhere in any of the briefing. But what he didn't do is, he didn't dispute the critical undisputed fact. The inventors, the PTO, everyone realized had this covered mixed data. That was the basis on which Chong served as prior art. That was the basis for the anticipation rejection. And it was overcome not because this claim arbitrarily blinds scientists to considering supplemental abnormal data. It was overcome because Chong did not reference registries.

So one of the things that counsel said was, "We have an 02 MICRO issue." He is right. We do have an 02 MICRO issue. The 02 MICRO issue is that counsel wants to import a negative limitation into the claim excluding blinding the use of supplemental abnormal data. We agree that issue should be addressed in this claim construction process. And the Court should conclude that that negative limitation is improper.

The second question, though, particular question is, are we advancing the ball in any way by changing normal to typical, attack-free, even though the inventors did not use those phrases in the claims. We are not.

Let's talk about Symantec's new indefiniteness argument, an argument that doesn't appear in the 30 pages of their opening brief, does not appear in the 30 pages of their responsive brief. It is a new argument that is designed to reflect, I think, a real basic misunderstanding of what the patent is about. If you remember from my presentation, I spoke a lot about Question 1 and Question 2. Question 1, what is the model; Question 2, what data do you use to construct the model. And I think the problem is, Symantec is saying we don't know how much normal data you need to use. In fact, I think I got their quote pretty accurately. "Is it one percent abnormal data that you have to use to construct your model or is it 100 percent abnormal data you have to use to construct your model?" What the patent says is you must base the model on records of plurality of processes that access the operating system and that are indicative of normal computer system usage. That's what you must have. It doesn't say that's the only thing you have; it is what you must have. And it is plural, so under the patent law, you would need to have multiple normal process access data points in the construction of your model to satisfy it. But that relates to what the data is used. I don't hear Symantec saying that it would be shocking if they thought the word "normal" was indefinite or what model of normal behavior was indefinite. A model of normal behavior requires you to construct an understanding of what is normal and how it is different from abnormal, just like the apple and the oranges.

Let's go to Slide 28 of Symantec's presentation, actually. If you look at Slide 28 of Symantec's presentation, they point to a portion of the specification that says, "If a model of normal registry behavior is trained over clean data, certain things happened." Notice the contingent: "If." Notice the reference to "clean data." What doesn't appear in the claims of the patent? The word phrase "clean data" does not appear in the terms of the claims of the patent.

Counsel spoke about the -- in my presentation, I gave two very clear definitive instances in which mixed data was used. I talked about the fact that they adjusted the model to take into account the fact that malicious programs are silent when they add, and they also talk about threshold adjustment. I heard nothing whatsoever from counsel about the program addition, use of malicious data, and I think the reason for that is because it is quite clear that program addition data that's used to construct the model did in fact involve malicious behavior.

As to the threshold point, what he said was, "Well, that's not about creating the model; that's about training the model or adjusting the model or making the model how you want it to be." Well, the claims don't say you have to train the model using normal behavior. Let me see if I have that right. He said it relates to creating the model. But the claims don't say, "Well, you have to use normal data to create the model." They use a different phrase. They just say, "The model must be based on X." I think that makes a lot of sense. If you look at the claim, the claim is agnostic in terms of everything you will use. It tells you what you have to use. You have to use registry system access information. That's representative of normal. It doesn't exclude using other type of information, so Symantec could just as easily come up and say, "We want a construction that says, 'You can only use registry access information.'" But that would make no sense. Because that language in yellow highlighting isn't limiting what you use. It is describing what you have to use.

Let's turn to the discussion of the '342 application. Can I have Slide 62, please? So one of the things that I think is important is to read documents in context. And it is a frustration of oral argument that sometimes when you show up a slide, you show up a quote, you don't really get the entire context. I'm going to try to give some context in the '342 application. These are not slides that were shown by counsel or passages that were shown by counsel.

One of the things that the application speaks about is it says, "Most of these anomaly detection algorithms in the prior art require that the data used for training is purely normal." Two important points: "Most," not all. And it is talking about the prior art. It is talking about what came before the work of the inventors. What's interesting, of course, is what words don't appear in the claims. "Purely normal" does not appear in the claims. The inventors clearly knew how to say, "We know about systems that use purely normal data." They didn't put it in the claims.

Let's go to Slide 63. What the application does say is that "We think it is valuable to include mixed data, data that includes information on intrusions, as well as information on normal processes." And this is a passage from Paragraph 106 that Symantec doesn't engage in its briefs, doesn't show in its presentation. It is an example of using an using anomaly detection algorithms that involve mixed data. Here is what Symantec says. They say, "There is a difference between supervised and unsupervised anomaly detections." So let's go back to our metric. What words appear in the claim. We talked about the fact that "100 percent clean data" doesn't appear in the patent. Guess what else doesn't appear in the patent? "Supervised." The claims aren't limited to supervised anomaly detection system. They know how to write "supervised anomaly detection system" and they didn't write it into the claims. Instead, they incorporate an application that doesn't just suggest you can use supplemental abnormal data, it describes a robust algorithm that allows you to use it.

One of the things Symantec says when they talk about the discussion of misuse algorithms in the '342 application, that that's criticizing that system for being computationally defective or computationally deficient. Those phrases, of course, don't appear in Paragraph 102 of the application that Symantec showed you. If you look at the '084 patent, they are criticizing misuse detectors that focus exclusively on abnormal data. That's what they are dissatisfied with.

Let's go to Symantec's Slide 34. I know, Your Honor, you may have it in front of you. I don't have it on the screen. So, to me, this is pretty neat. And it goes back to something we said. Symantec shows this section from this article that became the provisional application for the '084 patent and they quote language saying, "We used only attack-free data for training." They said, "Aha, see, they say in our experiments, we used only attack-free data for training." Interesting point. Two, in fact. Guess what phrase doesn't appear in the claims? "Attack-free data for training." Second interesting point, and why don't we go to our slide, 37. In that same article, they said, "We use very weak algorithms. We don't use sophisticated algorithms here. We don't use algorithms that can handle mixed data." Which they disclose in the '342 application, for example. And so to say -- what they are saying here is, "More sophisticated algorithms can be used, because algorithms do exist in the art, that reference that application, have the ability to handle mixed data." Of course, this article doesn't discuss mixed data, and the reason it doesn't discuss mixed data is because the algorithm they happened to be using to perform their experiments didn't have that capability.

So Symantec Slide 36 discusses a paper by Dr. Eskin. And they quote a passage where they say, "If there is an intrusion hidden in the training data, the anomaly detection method will assume that it is normal and not detect subsequent occurrences." They say, Your Honor, that this proves that you can't used mixed data in anomaly detection. But I think it is worth going to our rebuttal

Slide 1. What's neat about that quote is that they actually cropped it, because what the quote says in full context is, "There's these typical approaches that require training over clean data, normal data containing no anomalies." Once again, notice what doesn't occur in the claims. Clean data and no anomalies. They didn't show this passage of the article to you. What Dr. Eskin is saying there is, "There's these junky old methods that only use clean data. And now I'm going to describe a method of detecting anomalies, anomaly detection," which is what is claimed in the '084 patent, "that doesn't require the use of clean data."

Once again, the frustration of oral argument is that sometimes when slides are shown, you don't see the complete context of the claim.

So the last thing I'd like to do is, why don't we go to the '115. So one of the things that Symantec's counsel said, and I think I got it close, but forgive me if it is not exact, is "normal data" is used consistently to mean "clean data." Two interesting points. One, the phrase "normal data" doesn't appear in the claims. What the claims say is we want you to include registry access information that is normal -- that is representative of normal processes. So the phrase "normal data" doesn't appear, so even if "normal data" did mean 100 percent clean data, that would say nothing about the claims.

But here is a more interesting point. So Symantec is very focused on provisional applications and articles the inventors wrote and using designs in those articles to try to limit the claims. But in the provisional application for the '115 patent, which also references anomaly, that provisional application actually attaches a design document in which a model of normal data is created, and that model of normal data, guess what it uses? "It can include good data, potentially harmful data, and noise." This is the '115 provisional application.

Now, there was some confusion in that Symantec submitted the '115 provisional application in its briefing. But what we have noticed, and it was not clear to us why this was the case, was Symantec actually excluded this portion of the provisional application from the exhibit. So what we have done, Your Honor, is we filed a supplemental motion to Your Honor seeking to add to the record the complete provisional application. I believe this was Exhibit 12, if that's correct, to the Symantec brief. So we filed a new version of Exhibit 12 which includes the omitted portion of the provisional application, making clear that you can create a model of normal data that involves mixed datasets.

Symantec's last point is, once again, a new argument. It is not an argument you hear in their briefing. They want "based" to mean something different than "employed." Well, I wish they would have told us that so we could have briefed it as opposed to having to do it here without the benefit of full analysis and information. But from what I can tell, what they really are trying to do is to say, "Based on" means you can only use a probability to make your decision. And I think that that is the implication of what they mean by "based on." So once again, that would be a negative limitation, saying, "In your probabilistic model, you have to use probability and you can't use anything else whatsoever to make your decision. And we want you to import that negative limitation in the claim by using this phrase 'based on.'"

And I'm grateful that now we do know what they mean by "based on" so we can address it. But once again, there is nothing in the specification, nothing in the claims, that say you can't consider other information beyond probability. What the claims say, what the specification says, is that you must consider probability.

So with that, Your Honor, I'll now turn to the "anomaly" term, which will be very, very short. It is the next term.

THE COURT: All right.

MR. SHEASBY: So the term "anomaly," the dispute regarding "anomaly" is relatively constrained, and it repeats many of the same arguments that we have heard regarding the discussion of "normal." That's on 67. So -- let's go to 68. So the dispute regarding what "anomaly" means really recapitulates this desire to have a "model of typical, attack-free" imported into the definition of "anomaly." And the reason why they want to import that language into the definition of "anomaly" has nothing to do with how "anomaly" is described in the specification. An anomaly is deviation from normal behavior which may correspond to an attack. That is consistently how the phrase is used in the specification. In fact, the definition of "anomaly" that we have proposed is absolutely consistent with the specification as well. In the sense that -- excuse me, it is absolutely consistent with the extrinsic record. If you look at the definition of "anomaly" in the extrinsic record, it also speaks about deviation from models of normal.

So why do they want to put in the phrase "model of typical, attack-free behavior" into the definition of anomaly? Well, the reason why they want to do that is because we know now that they want to take the phrase "typical, attack-free," which they say defines the model, and use it to import a negative limitation about the data that you can include. You must blind yourself, without doubt, 100 percent, to anything other than purely normal data. Even though, and the reason they want to do that is because the phrase "anomaly" occurs in a different patent. It also a course in the '115 patent. And so they stick the concept of "typical, attack-free" into the definition of "anomaly" from the '084 patent. They are then going to use that to import the same limitation, negative limitation, you must blind yourself to any data other than 100 percent normal, pure data so that they can have that limitation in the '115 patent as well. And we can talk about that issue later today when we discuss the '115 patent.

I think that's what I have to say on "anomaly," Your Honor.

THE COURT: All right.

MR. NELSON: So, Your Honor, I'm going to resist the urge to go ahead, I believe what you heard was trying to respond point by point to what I had said. I said what I said, and I think Your Honor heard what I said. Me going back through and articulating those things again and trying to use the Court's time, we can't be here forever going back point/counterpoint, so please don't take my silence as the fact that I agree with anything, but I recognize as a lawyer that at some point we have to stop talking, even though I only used less than half the time of my co-counsel. But I think I'll try to put that in the bank and use that maybe down the road one time, Your Honor.

So just get to the point with "anomaly." "Anomaly," sure, we have the issue because we are comparing to the "normal computer system usage," so we have the definition of "normal computer system usage." There is no question about that. But I don't need to rearticulate thsoe arguments. That's what you are doing in the context of the claim. And if we look at Slide 46, Your Honor, you see as an example, and counsel went through these steps, you generate this probabilistic model of normal computer system usage, and then you analyze features from the record of a process that accesses the operating system registry to detect deviations from normal computer system usage to determine whether the access to the operating system is an anomaly.

So if we look at the next slide, we can see, and this is consistent with what everybody has talked about throughout in describing these inventions, is that here it is an object of the invention to generate a model, and then the model is used by the anomaly detector to decide whether each new registry access should be considered anomalous. So the way you determine an "anomaly," according to the claim language, and according to the description of the invention that we have all agreed with, is by making a comparison of the observed behavior, the currently observed behavior, to that model of normal computer system usage. That's why we have the model in there.

Now, Columbia's construction just omits that. So you don't, according to them, even the way we have all described this, in order to detect something is an anomaly, you don't even need to use the model that you generated. According to their construction, they leave it out. It is just any behavior that deviates from normal. So the model is something that you generate, according to them in their claim constructions, and then never use again, apparently, or at least you don't have to use it again. That's not the way this invention is described. So that's the reason why we have the construction that incorporates the idea that it is a deviation from that model of normal computer system usage, because that's the way the invention is described, that's the structure of the claim, and that's the way it is consistently used; therefore, it should be construed that way, not some other random, undescribed determination of whether something is normal or not normal, Your Honor. So that's the reason for the construction. And I don't have anything more than that on "anomaly," Your Honor.

THE COURT: All right. Thank you.

MR. SHEASBY: Your Honor, we have one more term for the '084 patent.

THE COURT: All right.

MR. SHEASBY: Your Honor, the next term in the '084 patent is "operating system registry." The essence of the dispute between the parties regarding "operating system registry" is that Columbia is trying to define the nature of the unique operating system registry, that unique structure in the computer that is the focus of the specification. Symantec is proposing a broad, generic definition which would include not just the operating system registry, but basically any file system that exists. At least, I think that's the intent of the construction. Even though the specification makes clear that the operating system registry is a unique species with unique features distinct from the file system.

So this is what the specification says. It speaks about the fact in the summary that its focus is "to generate a model of normal access to the Windows Registry." There is no artifice here. It is the Windows registry. They are fixated with it. They are so fixated with it that they actually define what they mean by "registry." They pull no punches. They say, "Well, you can create the system monitors programs to access the file system of the computer. One example of the file system is, of course, the Microsoft Windows Registry, hereinafter referred to as the Windows Registry or the Registry. The Microsoft Windows TM Registry is the registry."

So the answer to the question, What is the "operating system registry" that's referred to in the '084 patent? It is actually not difficult. In other words, this should be one of the easier terms in the day to address. It is the Windows Registry. What we tried to do, and apparently I think that is what has caused this chaos to begin with, is as opposed to saying just the Windows Registry, we ended up trying to identify for your Court what are the key features that distinguish the Windows Registry from just a generic file system.

And what I think should resolve this is the extrinsic record. If you look at how folks in the industry define the Windows Registry and how they make clear that the Windows Registry is distinct from generic file systems, they consistently say, "The Windows Registry is a hierarchical database. It contains keys and values." If you see our proposed construction, that's exactly what our construction says is the Windows Registry. Microsoft itself says what its registry is. "The registry is a hierarchical database. The data is structured in tree format. It has keys and values." The essence of Columbia's construction is these features.

Now, what Symantec does, it has two strategies. First, it says, "Well, if you are going to agree with Columbia, we want you to take everything in that paragraph, just shove it all into the definition of Windows Registry." This is a paragraph from the specification that discusses the Windows Registry. But the problem with that is that various portions of that paragraph discuss things that may be present, may not be present, that are generally present. The touchstones, what's always present, is that it has configuration information, it is hierarchical, and it has keys and values. So we are very uncomfortable with the strategy of putting all the information that paragraph because it doesn't necessarily reflect things that necessarily are always going to be present, except for keys, values, and hierarchy. Here is an example of that structure. Here is the tree on the left-hand side. On the right-hand side is the keys and the values.

So one of the things the specification makes crystal clear is that there is generic things known as file systems, but file systems are different than registries. Registries are unique. And so the specification says, for example, LINUX and UNIX systems have file systems and you may be able to extract some important information by looking at that. But that's not a registry. They are definitive about that. A file system is not a registry. A file system is a broader concept. What it really is, it is a Russian doll. The file system is the broad generic term, and within that is the concept of an operating system registry.

So the challenge that I have with Symantec's construction, and it may be that there's just ships passing in the night, is when you propose a database of information about a computer's configuration, aren't you really just merging together the concept of files and operating systems? The reason why I say that is because I don't think it is disputed, LINUX and UNIX file systems have configuration data in them. That's undisputed. So what Symantec's construction is doing is it is not showing fidelity to the specification.

So Symantec has this argument that it makes, and it makes it repeatedly, where we are just asking for the moon. We want these claims to be completely untrammeled. But the truth is, we are not asking for a broad construction, we are not asking for a narrow construction. We are just asking for the right construction. This is what it says. "The essence of a Windows Registry is hierarchy, trees, keys, and values." And that's what we are trying to show fidelity to. Thank you, Your Honor.

THE COURT: All right. Response?

MR. NELSON: All right. Yes, Slide 49. So here is the problem with this, Your Honor. I think you just heard it. So they want it to be -- the definition basically to be the Windows Registry. Right? That's it. The Windows Registry. They would like that to be the case. Of course that's not what the claim says. We have heard that, tried to turn it around on us 500 times, you know, that that's not in the claim. So the claim is not limited to that.

But there is an even bigger problem than that. Because, look, this claim, this patent, this provisional goes back to 2002, the application that led directly to the '084, for example, was filed in 2003. Well, Windows changed over time. We know they bring out new operating systems, completely revamp things, tell us Windows 7 is way different than Windows XP and some of the versions that came before that. So how do we know that the Windows Registry hasn't changed over time? Right?

So they want to hedge their bets. They want to say, "Well, it is just kind of this general stuff." So we described in the patent, and that was our problem with this, if we go to Slide 50, what we did is, we took the topical sentence, which is the general statement of what a registry is, "As is known in the art, the registry is a database of information about a computer's configuration."

Now, this section then goes on to describe a number of other things that are specific about the Windows Registry, not a general statement, but specific about the Windows Registry. "Registry contains information that's continually referenced by many different programs." If we go to the next slide, we see, this comes from something that Columbia wants included, "It is organized hierarchically as a tree. Each entry in the registry is called a key and has an associated value." But if we go to the next slide, we see here is another, you know, statement equally, "The registry is also the storage location for all security information such as security policies, user names, and passwords." So these are all specific things, properties, that the Windows Registry at the time this patent is filed has.

So now, but they don't want that, because the Windows Registry changes over time. So now they are going to come in and say, "Wait a minute, Windows changed its registry, now it is not the storage location for all security information such as security policies," so they say, "Oh, that doesn't matter, it is still a key." So they want to have their cake and eat it, or eat it and have it, or however you want to say that phrase. Because they want it to be very limited. They don't want the general statement of what a registry is, the topical sentence, that it is the database of information about a computer's configuration. Rather, they want it to be, well, specific to Windows, but since we know that registry has changed over time, it is only specific about certain things that that Windows Registry had. So that's the game that they are playing here. And that's the problem that we have with the definition.

So that's why we think it is helpful, it is the topical sentence in that paragraph, a general statement about what the registry is. These other things are specifics. If we are going to include one of the specifics about the Windows Registry, remember, as of 2003 when this patent is filed, if they want it to be specific to that, because there is no way they can claim Windows Registry in 2005 because they don't even know what it is at the time the patent is filed, so there is no way that could be described, then let's include everything. That's our point. You want to have it limited to Windows Registry, then let's freeze it in time the way it should be and have everything that you described about that. That's not what we are trying to do. We have tried to provide this look at the general definitional statement.

As to that's the problem with it. You can't have it both ways on this construction, Your Honor. And we think, that's exactly what Columbia is trying to do here.

THE COURT: All right. Any brief rebuttal on this point?

MR. SHEASBY: Very brief. Your Honor, I don't think this is game playing. In other words, the purpose of claim construction is to struggle with an issue, what is a registry, what are the essence -- what is the essence that makes a registry a registry. And in my mind, the answer to this question is not something that needs to be debated by either you or -- either myself or my colleague telling you what a registry is or what's important in a registry. It can be answered by what's in the record. And what the record consistently says, the references, the patent: Hierarchy, keys, and values. And that's what we have tried to reflect in the specification. Thank you, Your Honor.

THE COURT: All right. I think we should stop now for lunch. And if you all could come back, let's make it 2:15, we will get started with the afternoon session.

(Luncheon recess taken from 12:56 p.m. to 2:15 p.m.)

THE COURT: All right. Let's move on to the third family of patents.

MR. SNYDER: I have copies of slides for the Court.

THE COURT: All right.

MR. SNYDER: Your Honor, the '115 and '322 patents are the last family of patents at issue in this case. The '115 is the parent patent and the '322 patent is the continuation. They are both entitled "Methods, Media, and Systems for Detecting Anomalous Program Executions" and they relate to work by Columbia Professors Stolfo and Keromytis and their graduate student, Stelios Sidiroglou.

The main problem that the '115 and '322 patents are directed to is that sometimes malicious programs cannot be detected until they are run. When we talked earlier today about the '544 and '907 patents, there was the term "static analysis" or "static features." Those are attributes of a file that you can discern without executing the file or the program. This patent family is talking about something different, attributes that you can only discern if you actually run a program.

The way that you actually gain insight into the program behavior is you look at something called function calls. What's a function call? Well, when somebody is writing a piece of software, they don't want to reinvent the wheel every time. For example, if they want to talk across the network and they don't reimplement the TCP/IP network stack. Instead, they use code that other people have already written to do common functions. This can be writing a file, this can be talking across the network, it can be creating a user interface window. All of those are traceable to function calls. So they give insight --

THE COURT: Excuse me. (Court conferring with Clerk.) Go ahead, I'm sorry.

MR. SNYDER: Because the important things that a program do correspond to the function calls. If you track and analyze the function calls that are made by a program, you can gain insight into its behavior, and that's what these patents are about. Here is Claim 1 of the '115 atent and its exemplary. It is "A method for detecting anomalous program executions, comprising: Executing at least a part of a program in an emulator." "Emulator" is one of the terms that's up for construction today.

The next element is "Comparing a function call made in the emulator to a model of function calls for the at least a part of the program." This shows how the emulator is being used. You are executing the program in the emulator, and it gives you visibility into the function call. That is, the emulator is the component that actually let's you see what function calls are being made. If you don't have an emulator or you are just executing a program regularly on your own computer, you can see some things that it does like if it pops up a window, but you don't normally understand the function call that's being made. You have to have another software component or program that's running alongside the main program that lets you actually inspect the function calls.

The next element is "Identifying the function call as anomalous based on the comparison." "Anomalous" is another claim term at dispute. This is when you detect that the function call is indicative of doing something bad. The program is a virus or the program has been infected. Then "Upon identifying the anomalous function call, notifying an application community that includes a plurality of computers of the anomalous function call." This is what you do after you discern that the program might be malicious or it is exhibiting anomalous behavior. An "application community" here is members of a community running the same program.

So there are three terms in dispute in this patent family. I'll be discussing the emulator term and my colleague, Jason Sheasby, will be discussing the other two terms.

Columbia's construction of emulator is "Software, alone or in combination with hardware, that permits the monitoring and selective execution of certain parts, or all, of a program." Symantec's construction is, "Software, alone or in combination with hardware, that simulates a computer system."

And in many respects, the dispute about this term is a dispute about which evidence to look to. Columbia's construction is grounded in the specification and how the specification describes the emulator. Symantec's construction, on the other hand, is mostly based on extrinsic evidence and one particular embodiment in the specification. Just going by the PHILLIPS hierarchy, Columbia's construction adheres most truly to the proper meaning of the term.

There are two questions that need to be answered with respect to the term emulator. The first is, what is an emulator in the patents. Here, both Columbia and Symantec agree that it is software, alone or in combination, with hardware. The second question is where there is a dispute. It is, what is the role of an emulator in the patents? Symantec says that it simulates a computer system and Columbia says it permits the monitoring and selective execution of certain parts, or all, of a program.

Columbia's construction is based in the role of the emulator as described in the specification. The emulator is described as software that operates alone or in combination with hardware, in Column 13. In numerous places in the specification the emulator is described as monitoring another program. Furthermore, all or selected parts of the program may be monitored. Lastly, the emulator permits the selective execution of the monitored program. And all of these attributes are replete throughout the specification and applied to every single disclosed embodiment.

Here are some citations for the monitoring functionality. It shows "an application that monitors other applications." This is how you get insight into the function calls that the program is making. Another citation says that you are "monitoring and analyzing application-level program function calls." Another cite is "The use of an emulator allows the system to detect and/or monitor a wide array of software failures." So you are looking at another program and trying to gain insight into what it is doing by examining the other program's function calls.

The other key aspect is selective execution. Using an example from one embodiment, called STEM, it was a program that was developed at Columbia, and it interoperates with something called the Valgrind emulator. And yes, I had to look that up. STEM is described as "permitting the selective execution of certain parts, or all, of a program."

Symantec criticizes Columbia for supposedly importing the selective execution limitation from the STEM embodiment into the overall definition of emulator. But that's simply not correct. Every single disclosed embodiment in the specification also does selective execution. Furthermore, the patent makes pains to say that "any other suitable technique for emulation can be used." It is trying to avoid being limited to one particular narrow version of emulator.

The important part of the specification that is crucial to the claim construction discussion for emulator is Columns 13 through 16 of the '115 patent specification. So it is four columns, it is two pages. There's discussion earlier on in the specification about an emulator, and also a discussion afterwards. But this is where the patentee really digs into what an emulator is and what it can do. It also discloses several different embodiments of an emulator. There is an embodiment that is "an instrumented version of an application." This is at the top of Column 13. There is another that utilizes something called a sandbox, which we will discuss later. That's towards the bottom of Column 13. There are embodiments that are compiled into the code or linked at Column 14. And there is also embodiment that is invoked in a manner similar to a debugger at Column 14, 10 through 15.

Every single embodiment here does monitoring and selective execution, but the embodiments differ. If you read these columns of the specification, the impression that you get is that the patentee was trying to encompass several different varieties of emulator. They weren't limited to a simulator, they weren't limited to a sandbox, and they weren't limited to any other embodiments. They were claiming multiple embodiments.

What's the problem with Symantec's construction? It improperly replaces "emulate" with "simulate." This is in consistent with the specification because it excludes several embodiments. It is also not helpful to a jury. If you take that "emulator" is not a term that a layperson would really know and replace it with "simulator" or "simulates," you are not really providing any further clarity to the jury. You are simply guaranteeing there's going to be another dispute down the road between Columbia and Symantec about what it means to emulate a computer system. That is not helpful to the jury and is not the right way to do claim construction.

Simulation is explicitly not emulation. There are three particular reasons I'm going to go into today. The first is that simulation as described in the patents in the way the patents use it is "an optional feature that occurs after an anomaly has been detected." Let's dig into that a little bit more. Here is a portion of the specification at Column 15, and there is a red box and a blue box. In the red box, it describes an anomaly already having happened. So you monitor for failures prior to executing instructions, you can revert memory changes, and store memory modifications. Then you can do something else after you have detected an anomalous function call, and that's the blue box. After you have detected the anomalous function call, you can also simulate an error return from the function. This is sometimes referred to herein as error virtualization. This is how the patents talk about "simulation." The word "simulator" never occurs in the specification or the claims. The only occurrence is the word "simulate." It happens twice, one in Column 13, one in Column 15. And in both situations, it is describing the error virtualization feature. So if you are going by the patent's own language, when it talks about simulation, it is talking about error virtualization.

The interesting thing here is that there are dependent claims that claim error virtualization. Here it is generating a virtualized error in Dependent Claim 5. This claim depends on Independent Claim 1, which we saw a little bit earlier. Claim 1 has "emulator," but it doesn't disclose error virtualization. You have to go all the way to the dependent claim. This is a textbook example of importing an optional embodiment into all the claims. This is explicitly a dependent claim if you go by the patent's own interpretation of what it means to simulate.

Another interesting thing here is that Columbia raised this issue twice. We raised it in our opening brief, we also raised it in our reply brief. And Symantec never addresses this. They don't have any explanation for why this error virtualization feature isn't how the patent is considering simulation.

There is another important reason why simulation is not emulation. It is that the disclosed emulator embodiments are not quote/unquote fake. Where is this word "fake" coming from? It is coming from the extra gloss that Symantec is trying to put onto their own construction. This is Symantec's opening brief. "Those of ordinary skill at the time would have considered the plain and ordinary meaning of emulator to require program execution that is fake or simulated." This is Symantec's expert, Dr. Ford. This proposed construction omits what I consider to be the key requirement of an emulator, that it executes a program in a manner that is fake or simulated, not real. These extra interpretations or glosses on top of their construction aren't helpful. We are going to argue over what it means to be fake or not real.

Furthermore, even taking them at their word that this is what stimulation means, that it implies that you are executing a program in a manner that is fake or somehow not real, what this most closely approximates is one particular embodiment in the '115 patent. This is at Column 14 at Lines 47 through 50. It is describing something called a "sandbox." "A sandbox generally creates an environment in which there are strict limitations on which system resources the instrumented application or a function of the application may request or access."

You will notice it talks about the instrumented application. An instrumented application itself is only one particular embodiment, and this quote is talking about putting the instrumented application into a sandbox. So it is one variation of one particular embodiment. Now, what does a sandbox do? It says, "It creates an environment in which there are strict limitations on which system resources the instrumented application or a function of the application may request or access." So the idea of a sandbox is trying to get to is you don't want to let the program interact directly with the computer system or the operating system. For example, you don't want to let the program create a real file on the hard drive, or if the program is trying to make a network transaction or a network request. You don't want them to actually make a real request. You just want to give them some fake information that makes them believe that they have made a real request, but they actually haven't.

So that's what this sandbox embodiment is talking about and that's the closest thing you can actually find in the specification to Symantec's idea that the program execution has to be fake or not real on the cited emulator. The problem is a sandbox is just one particular embodiment. It doesn't describe the other embodiments. In fact, there are three specific embodiments that are exclusively not fake and execute on a real operating system and interact with a real computer. The three embodiments are "debugging functionality," at Column 14, Lines 6 through 15; "instrumented code," Column 13, 3 through 13, and a "compiled instruction-level emulator" at Column 14, Line 45 through Column 15, Line 5. Dr. Szajda discusses in his second Declaration at Paragraphs 22 through 23 each example of these emulators and explains why they are interacting with a real computer and are not executing in a fake manner.

Let's talk about the debugging functionality embodiment. This is what the specification says. "In another suitable embodiment the instruction-level emulator may be invoked in a manner similar to a modern debugger when a particular program instruction is executed." What's a debugger? It is a tool for software programmers to use when they are writing a new program. They write some source code, they have a working version of the program, and they want to run their intermediate version and see how it behaves. So they run the program that they are developing, and they also attach a second program to it called a debugger. The debugger lets the programmer see what the program that they were writing is doing. It lets them step through the instructions one by one in the debugged program.

This portion of the specification is saying that you can use an emulator in a manner similar to a debugger where you have a regular program that is executing, and then you attach the emulator to it, you look inside at the other program and see what it is doing, and you can selectively execute it. There is nothing about a debugger that prevents the debugged program from affecting the real computer system. It is not fake. It is not simulated.

Here is a particular source that was cited in Columbia's briefing. This is from a description of a program called GDB, which is a well-known debugger in

LINUX/UNIX systems. It says, "The purpose of a debugger such as GDB is to allow you to see what is going on inside another program while it executes or what another program was doing at the moment it crashed." Professor Szajda explains in his Declaration that this is describing real execution, not fake execution.

There is another embodiment that is also real and not fake. It is the instrumented code embodiment described at the top of Column 13. The specification says, "The system may generate an instrumented version of the application. For example, an instrumented version of the application may be a copy of a portion of the application's code or all of the applications's code. The system may observe instrumented portions of the application."

So what this is talking about here is that you have a program that already exists. It could execute outside an emulator, but you are deciding to put it inside an emulator. And you copy the actual program code so you are still executing the actual program code. The actual program code is still directly affecting the operating system. This is not a fake execution.

There is a third embodiment, which is an instruction-level emulator, and there are two examples at the top of Column 14. You can link it with an application or you can compile it into the code. When the specification is talking about linking and compiling here, it is talking about different ways to integrate two different programs together. When you are doing linking, you are talking about making a connection or a reference between two separate program files. When you are talking about compiling into the code, you are talking about combining two different program files into one so you have a chimera program.

Both of these still have the regular application executing, not in a fake environment. The program is actually making real transactions and talking to the operating system just like a program that would not execute in an emulator is doing.

There is a third major reason why simulation is not emulation. It is that Symantec's extrinsic evidence contradicts its claim construction. If you look at Symantec's briefing, they have about a page of different extrinsic evidence sources. So much of their rationale for their construction is from extrinsic evidence which they claim defines what an emulator is. The problem is, their definitions don't say what Symantec is saying they say. Here is one definition that's from Symantec's brief. It is the Dictionary of Computer Science Engineering and Technology. What you see on the screen here in Slide 33 is the definition that Symantec gave in their brief. "(1), the firmwear that simulates a given machine architecture." So there is an immediate question here. What is this term "machine architecture" even doing here? In our case, in this patent, we are talking about emulating all or part of a program. Where is this term "machine architecture" coming from? It seems like maybe this is for emulator in a different context.

The other thing is that Symantec leaves off the second definition for emulator. The second definition doesn't mention simulation. In fact, it ends with "compare with simulator." So this definition that they used is saying there are some types of emulators that are not simulators. It is as simple as that. The same dictionary, when it is defining the term "emulation" says, "Contrast with simulation." This is an even stronger signal than "compare." It says, "Contrast with simulation." Their own definitions are not supporting their own claim construction.

Here is a dictionary definition that Columbia cited. The McGraw-Hill Electronics Dictionary. It says that "Emulation should be distinguished from simulation." So even given all this effort that they put into their essentially wholly random extrinsic evidence definitions together to try and support their claim construction, the extrinsic evidence doesn't really support them. There are emulators that are not simulators. They are just cherry picking the extrinsic evidence. What this really comes down to is the specification. If you look at Columns 13 through 16, you will see that the specification defines different types of emulators and describes them all as doing monitoring and selective execution.

Symantec's position is mainly extrinsic evidence. They also talk about something called a virtual processor. Virtual processor, even if it is a limitation, is only a limitation of one embodiment, which is the instruction-level emulator. Further, they don't even link virtual processor to the concept of simulation. They just say, "Look, it says 'virtual processor' in the specification. This is talking about simulation." But they don't really explain it.

Columbia's construction adheres to the specification and describes what the emulator is actually doing in the context of the claims.

THE COURT: All right, we are going to have to take a break. I have to deal with this jury. We will take a break until we get set up on this other case and then we will get back to you guys.

(Recess taken from 2:41 p.m. to 3:20 p.m.)

MR. HAMSTRA: Your Honor, Nathan Hamstra for

Symantec Corporation.

THE COURT: All right.

MR. HAMSTRA: Go to Slide 56. So in the claims of the '115 and '322 patents, the emulator is a particular structure that's recited in the claims. That particular structure has particular plain and ordinary meaning. An emulator is simply software that simulates a computer system. Just looking at the text of Claim 1, we see that Columbia's construction has a couple problems. So the first element of the claim reads "executing at least a part of a program in an emulator." But Columbia's construction of "emulator" is just something that allows for the execution of part or all of a program.

Similarly, the second limitation simply recites comparing a function call made in the emulator to a model. In other words, the second limitation is talking about monitoring or analyzing a function call. But again, Columbia's proposed construction of "emulator" is something that permits the monitoring of all or part of a program. So in other words, Columbia isn't trying to construe what an emulator is. Columbia is rather simply construing various uses an emulator is put to. And that's telling, because they aren't attempting to construe emulator here.

Now, the '115 and '322 patents use the term "emulator" consistent with the plain and ordinary meaning. Turn to Slide 57. So this portion of the specification here is discussing the emulation of a piece of code of a program. And it discusses how, when the emulation starts, execution switches to the program executing on the virtual processor. And then when it reaches to the end of the portion of the program that is being emulated, the execution changes from the virtual processor to the actual processor. And that process is completely transparent to the program being emulated. It doesn't know it is being emulated.

Here, this portion expressly contrasts the virtual processor of the emulator and the actual processor of the system, which is precisely the point of Symantec's claim construction. Now, Columbia quibbles a little bit about "virtual" versus "simulated," but Columbia really hasn't articulated any particular distinction between those terms. And what's clear from the weight of the evidence is that an emulator must be a simulated or virtual or fake environment. And therefore, an emulated program is a program that's running in that simulated or fake or virtual environment.

Next slide. Now, Columbia cited some prior art patents to the Patent Office during prosecution of the '115 and '322 patents. As we noted in our briefing, these patents that are cited to the PTO during the prosecution form part of the intrinsic record, and consistent with Symantec's proposed construction, this first patent, U.S. Patent Number 5,978,917, defines "'emulation' means running a computer program in a simulated environment." Similarly, the second patent cited here, U.S. Patent Number 6,952,776, describes an emulation step "that executes the current object," that is, the program, "in a virtual environment."

Next slide. Symantec also cited a bunch of treatises from the computer security art in its briefing, and these treatises are likewise consistent with Symantec's proposed construction. For instance, the first book is a book entitled The Art of Computer Virus Research and Defense, and it is discussing code emulation. For code emulation, it says, "A virtual machine is implemented to simulate the CPU." Similarly, the next book is a book about virus and antivirus software entitled Malicious Mobile Code. And this book talks about an emulation engine that loads a file into a protected area of memory and then simulates the computer's operating environment. The last citation here, Virus Bulletin, is a magazine devoted to computer security and particular virus software. And Virus Bulletin has a glossary, and their glossary defines emulation as "any method of creating a fake environment." So the '115 and '322 patents themselves, other intrinsic evidence and extrinsic evidence all consistently agree that an emulator is software that simulates a computer system.

Next slide, Jerry, 60. Now, Columbia takes much of their construction for the term "emulator" from this quote of the specification here. This is a citation to Columbia's brief, and they selectively quote a portion of the specification saying that STEM, which is a technique in the '115 and '322 patents, "permits the selective execution of certain parts or all of a program." And to arrive at their actual construction they just put "software" with "hardware" in front and then add "execution and monitoring" to this statement here.

But let's turn to the next slide and see what this portion of the specification states in general. What we see is the embodiment being discussed here, Selective Transactional EMulation, or STEM. STEM "permits the selective execution of certain parts, or all, of a program." But what Columbia omitted from that quotation from their brief is "inside an instruction-level emulator using the Valgrind emulator." In other words, this portion of the specification here isn't trying to define emulator at all. Rather, it is describing a use, a purpose for which an emulator is used. But just because an emulator is used for a particular purpose doesn't mean that everything that performs that particular purpose is an emulator. That's a basic error in logic.

In this embodiment here upon which Columbia relies so heavily, the Selective Transactional EMulation embodiment, we know what the emulator in this embodiment does, too. So this '289 provisional application was the application to which the '115 and '322 patents claim priority. And it was also incorporated by reference into the '115 and '322 patents. And as you see, the heading at the top of the quote here, it is discussing Selective Transactional EMulation or STEM. And what does it say about the emulator used by STEM? The highlighted portion here says, "the emulator snapshots the program state and executes all instructions on the virtual processor." So again, even the embodiment that Columbia is relying on recites an emulator as software that creates a virtual system.

Now, Columbia counsel mentioned a few embodiments in the specification that they described as emulators, and we just have some dispute with a few of those so I wanted to point that out. I don't have slides for these, so I apologize. But if you want, you can turn to slide -- okay, go ahead. They reference an instrumented version described at Column 13, Lines 3 through 13 of an application. That portion of the specification does not describe that instrumented application as an emulator. They reference a debugger at Column 14, Lines 10 through 15. Similarly, the specification does not describe the debugger as an emulator. Rather, they only say it may be invoked in a manner similar to an emulator. But that's not the same thing. All it means is that it is using processor capabilities that are built in there for debugging to begin the emulation process.

Lastly, Columbia referenced an emulator that is compiled into the program itself. And they say that that's not a fake program. But that emulator that's compiled into the code, that's still entirely consistent with Symantec's proposed construction because that emulator also creates a simulated computer system to allow for the execution of the program into which it is compiled in a virtual environment as well.

Thank you, Your Honor.

THE COURT: Thank you. Brief rebuttal?

MR. SNYDER: I'll be brief, Your Honor. This claim term really comes down to what sources of information you are going to use to arrive at the construction. Columbia's approach is to look at the specification, carefully read the common features of the embodiments, and derive a construction that's faithful to the specification and actually describes what the patentee was talking about. Symantec's approach is to use extrinsic evidence, a few random pieces of prior art patents, a few random external treatises, and say that these definitions should be imported into the claims when the specification only mentions "simulate" twice and only in the context of the error virtualization optional feature.

Could I have Columbia's slides, please? I just want to briefly run through some of the definitions that Symantec talked about. They cited Patent Number 6,952,776, and this patent says "a program emulation step that executes the current object in a virtual environment." This definition, so-called definition, doesn't mention emulator, and doesn't mention simulator. Instead, they point to virtual environment. But they never really explain why virtual environment or virtual processor, which is the term that's actually in the specification, maps to their claim.

Here is another piece of evidence they cite in their brief and was just up on a slide a few minutes ago. It is by the late Peter Szor. He was a Symantec antivirus researcher and worked at McAfee as well. He had a book, The Art of Computer Virus Research and Defense. They cite a section from this saying an emulation of "virtual machine is implemented to simulate the CPU and memory management systems to mimic the code execution. Thus, malicious code is simulated in the virtual machine of the scanner."

The problem with this is that the sentence immediately after Symantec's quoted sentences is some early methods of code emulation used debugger interfaces to trace the code using the processor. However, such a solution is not safe enough because the virus code can jump out of the emulated environment during analysis. We talked earlier about the debugger embodiment that's in Column 14 of the '115 specification. It is talking about using the same debugger interface that Symantec's own extrinsic evidence is saying doesn't count as genuine code emulation. They are talking about a different type of code emulation. So there is a conflict here. It is between the type of emulator that the specification sets up, which has specific roles in the context of the patents, and some abstract emulator that maybe works with Symantec systems. Who knows? It is trying to import extra limitations from the extrinsic evidence. So there are some problems with the extrinsic evidence they are citing.

Mr. Hamstra said there is a problem with

Columbia's construction which is that we are only describing what an emulator does and we are not defining what an emulator is. Their construction has the same problem. It says that it simulates a computer system. That's a function, just like monitoring, and just like selective execution is. Really, the choice is, do you describe how the emulator is actually being used in the specification and in the claims and its relevant features even if the specification doesn't have an explicit definition of emulator, which it doesn't, or do you use extrinsic evidence? And PHILLIPS v. AWH says you have to go with the specification. Thank you, Your Honor.

THE COURT: All right. Next term?

MR. SHEASBY: Good afternoon, Your Honor. Your Honor, the next term we need to discuss is "anomalous" as used in the '115 patent. If Your Honor remembers earlier today I pointed out that "anomaly" and "anomalous" is both a term that appears in the '084 patent as well as in the '115 patent. We spoke about Symantec's use of the phrase "model of typical, attack-free" and insertion of that into the definition of "anomalous" is so that could then be imported as a limitation into the '115 patent.

Now, one of the things that counsel said earlier today, and I think is interesting, he said we have to put in the reference to "model of typical, attack-free" into the definition of anomaly because otherwise no one will know how you test for whether there is an anomaly. What test do you use. Well, that doesn't make any sense. Because in the '084 patent, the specification or the claims are clear that you know there is a deviation from normal by comparing it to a model of normal behavior. So clearly in the '084 patent, that language is not necessary for the purpose that Symantec represents to Your Honor.

And it is not really necessary for the '115 claims, either. And the reason for that is that the '115 claims actually go into excruciating detail, excruciating detail about the model that they want to have constructed. I'm on Slide 39. They say, "The model you should use is a model of function calls for the at least a portion of the program, wherein the model is a combined model created from at least two models created at different times." So this is one of the claims in the family, and you will see the great, great detail that they use to describe the model you are supposed to use to detect the anomaly. And of course, they don't say, "typical, attack-free," they don't say, "a model that excludes any supplemental abnormal information." They don't say, "Blind yourself to the common standard techniques that have been used for years," which is to use supplemental abnormal data. They say nothing of the sort.

In fact, if you look at the Summary of the Invention, you will see no reference to using 100 percent normal data, you will see no reference to using clean data. All of the phrases, the buzzwords that Symantec points to in articles, related applications that they say establish when you must use 100 percent clean data, none of those phrases occur in the Summary of the Invention of this patent or in the '084 for that matter, and none of those phrases appear in the claims of either this family or the '085 patent.

So I know it is getting late in the day, and if you will allow me to do so, Your Honor, I'm going to skip a couple slides and get to what I think is a really fascinating and interesting point.

So one of the things that Symantec says in its reply brief, in the second brief, is it says, "The '115 patent relates to the creation of anomaly detectors. And anomaly detectors by their very nature detect divergence from normal. When you detect divergence from normal, you basically have to use 100 percent pure data." That's Symantec's position. I don't think I'm caricaturing it. I know that's a common argument to the technique. I think in fairness that's the issue they are asking you to decide in both the '084 patent family and the '115 patent. You can't detect divergence from normal by using datasets that include both normal and abnormal data. That's their basic proposition.

Well, what's challenging about that is it has no connection to the record before Your Honor. Let me give you an example. The patent claims in the '115 patent, the independent claim, I'm on Slide 43 now, says "a method for detecting anomalous program executions using a model of function calls." What Symantec's position is that once you say "anomalous," once you say "model of normal behavior," you are automatically in a realm in which you can only use 100 percent pure data. You must blind yourself, take out your eye to this massive set of information that you can use to enrich your datasets.

But that can't be right. Because the dependent claim makes clear that that model of at least a part of a program must, can include, not must, but can include attacks. This is a very important point. Because what it reflects is the exact opposite of what Symantec is representing to you as the common, ordinary understanding of anomaly, of detection of divergence from normal. What Symantec is saying it is impossible to detect divergence from normal using anything other than pure data.

And that could not be more incorrect. It renders the claims, the dependent claims of the patent, an absurdity. Symantec actually has an interesting argumentative move. It does this with Claim 7, but I think, I anticipate they will try to do it with Claim 8 as well. They will say, "Well, Claim 8 is just describing a situation in which you are referring to at least part of a model." But that's the narrowing limitation from Claim 1. But as you see, that doesn't work because Claim 1 also refers to at least part of a model. So I did want to flag that argument, because I don't think it holds up when you actually compare Claim 1 to both Claim 7 and Claim 8.

This is a differentiation issue. And the Federal Circuit is quite clear on this. It is the idea that there could be no cogent way in which anomaly detection, detection of departure from normal, could in its ordinary meaning exclude the use of supplemental abnormal data when the dependent claim specifies that the solution -- the consideration of abnormal data is an option.

I don't want to sell a bill of goods to Your Honor. In other words, there are many types of algorithms that are not sufficiently sophisticated to consider anything other than purely normal data. That's absolutely the case. In many ways it is a much less sophisticated algorithm. It makes it easier to experiment with and is an algorithm that the inventors actually used in many, many situations. But to say there are only algorithms in which you can use only 100 percent normal data is really missing the point because none of these claims are limited to a particular algorithm. In fact, if you read the specifications, they make that clear. The inventors are unabashed. "We used a very simple algorithm. You can use more complex ones."

Just one final point: Symantec once again focuses on a provisional application and represents to Your Honor that in these provisional applications, the models of normal behavior, the only way to do it is to use purely normal data. But we know that's not correct. Part of the provisional application they did not include makes clear you can create a model of normal data using mixed data.

At that point, Your Honor, I think I'm done with that section and I'll save time for a very brief rebuttal.

THE COURT: All right.

MR. HAMSTRA: Go to Slide 66. Looking at Claim 1 of the '115 patent in the context of the anomalous limitation, the claim first recites "comparing a function call made in the emulator to a model of function calls." Then it recites "identifying the function call as anomalous based on that comparison." So what we are doing here is we are measuring the anomalousness of the function call based on a comparison to this model. Symantec's proposed construction requiring that the anomalous function call be measured against the model of typical, attack-free computer system usages is consistent with both the intrinsic evidence and the extrinsic evidence. Much of this is a rehash of what we discussed in the '084 patent and '306 patent, so I'll be brief here.

Starting with the provisional application, we see that anomaly detection is a known technique in the art. Anomaly detection algorithms build models of normal behavior and use those models to detect behavior that deviates from normal. The next question is, what is a model of normal behavior. The intrinsic evidence confirms that a model of normal behavior is a model of typical, attack-free behavior.

The '115 and '322 patents themselves at Column 3 Lines 50 to 52 describe building the model from normal data. The embodiments discussed in the '115 and '322 patents are entirely silent about using any abnormal or attack data for modeling purposes.

Slide 68. The '289 application likewise confirms, consistent with the plain and ordinary meaning of the term "anomalous," that the training is done using attack-free records. And interestingly, the documents cited by Columbia at Slide 46 indicating that there is some non-normal data included in the model actually contradicts that statement. Because the document Columbia cited actually describes all that information as normal data. "The normal data can include good data, potentially harmful data, and noise." So again, it is consistent with Symantec's proposed construction.

Now, Columbia makes much of claim differentiation. The Federal Circuit has cautioned against an oversimplistic application of this principle, though. One Federal Circuit case in particular said that the doctrine of claim differentiation cannot broaden claims beyond their correct scope determined in light of the specification and the prosecution history, and any relevant extrinsic evidence. That's MULTIFORM DESICCANTS v. MEDZAM, 133 F.3d 1473. That caution should be taken to heart here, particularly where the disclosed embodiments don't describe a model that actually includes attack data. Here, the correct scope is Symantec's proposed construction, "a deviation from a model of attack-free typical computer system usage." Thank you.

MR. SHEASBY: Just a brief rebuttal and then one final term. So at lunch, I was looking over Symantec's slides. And this slide that they showed, which is Slide 68, was actually a new argument that I hadn't seen before. This is one of the pieces of the provisional application for the '115 application. And you see how they are referring to something called a "one class SVM system." It is a type of algorithm. They are saying in that algorithm they use attack-free records. What's interesting, of course, is the phrase "attack-free" doesn't appear in the claims of either the '115 or the '084 patent.

Let's have the next slide now. So the portion, what they are citing here is actually an appendix to the provisional application. And the appendix is a portion of the appendix which, actually, let me skip that slide, the portion of the appendix they cite to is Appendix B. They are citing to Appendix B to have a discussion of RAD, which is a type of software algorithm. And if you go to that portion -- if you actually read the article that they cite to, in context what the article says is that the OCSVM system and the PAD system are different. We have shown that the PAD system is more reliable.

Let me take this in pieces. So the article that they are relying on in this slide is Appendix B to the provisional application. The provisional application says, "Look at Appendix B because it is going to have an interesting discussion of PAD/RAD," which is a different type of algorithm. When you go and you look at the actual article, it says, "PAD is fabulous, OCSVM," the portion they are referring to which uses only attack-free data, "is not very good."

Why is this interesting? Well, the reason why it is interesting is because -- let me stop here for a moment. I think there are very limited instances in which inventor testimony is helpful during claim construction. I think it is extremely rare. But I think one instance in which it is relevant is when the actual experiments they do are put into issue. And what Symantec has been doing today is they have been saying, "Well, they have used algorithms to run their experiments that use 100 percent pure data, and so I want to import those into the claim and we are going to point to articles in which they were using these algorithms which could only support the use of 100 percent clear data."

And one of the things that happened earlier this month is they actually took the deposition of Inventor Hoenig. Inventor Hoenig actually works at Google now. He is completely independent. What I mean by that is, we don't interact with him, we didn't prepare him for his deposition. He actually just came in and spoke cold at his deposition. And one of the things he spoke about at his deposition was, there are different types of algorithms that the inventors were using. One of those algorithms that they were using is an algorithm called PAD. Why is that important? Because PAD is the algorithm that's referenced in the '115 application, and it is the algorithm that's referenced in the '084 application as potential candidates for use. Here is what he says about the PAD algorithm. He says: "In the PAD algorithm, we could build a model of normal behavior using mixed normal and abnormal data." I'm on Slide 6 of the supplemental slides. There wasn't a requirement to use 100 percent pure data.

So they had algorithms available to them, PAD, for example, that didn't require 100 percent pure data. He said, "I want to be open with you." These are unfortunately not in your slides, Your Honor, but I'll get you copies and I apologize for that, but it is an exhibit -- none of this is new evidence. I want to be clear. These are all lodged. These are excerpts from the deposition of Inventor Hoenig in Exhibit Z. What he says is, "I want to be clear, we had the PAD algorithm, which was -- allowed us to use normal data supplemented with abnormal data. We had other algorithms which didn't allow us to use supplemental data. We had to use 100 percent pure data. They were weaker algorithms, less complex. PHAD was one of them."

So PAD allowed the use of supplemental abnormal data. PHAD, P-H-A-D, did not allow the supplemental use of abnormal data. Why is this so important? Well, the reason why it is so important is because if you look at both the '115 patent and the '084 patent, '115 patent, Column 4, Lines 9 through 10, '084, Column 18, Lines 5 through 9, they both make clear that you can use PAD as an optional way of -- optional algorithm for doing your anomaly detection. Why is that so important? Because PAD is one of the algorithms that allows you to use both normal and abnormal data.

So we spent a lot of time today talking about this issue. And what makes it sort of frustrating, excruciating, is because there is actually a scientific answer behind this. In other words, you may ask yourself, "Well, Mr. Sheasby, if they were using -- doing experiments with algorithms that only used 100 percent pure data, why shouldn't I import that in the claim as a negative limitation?" Putting aside the Federal Circuit says no. In other words, it seems like the inventors are trying to get away with something. But they are not trying to get away with anything. They used very simple algorithms because it allowed them to do efficient, targeted experiments. But they also published and made clear that more robust algorithms, PAD, for example, could allow this mixed data.

And so that's why we keep coming back to the question of where Symantec is saying "The ordinary meaning of normal excludes the use of any supplemental abnormal data." It is really reflecting a state of a fact that doesn't connect to the science. And I think that's why in this very narrow situation, looking at what Mr. Hoenig said, completely unprompted, completely independently, is actually pretty relevant.

So with that, I have no more on "anomaly" and I will move on to the final term, Your Honor.

So the final term is "application community." And as is often the case, it is the last term in -- it is the last term in the brief and it was the last term today. And so sometimes when you are last, you get short shrift. Just ask people whose last name is Z in elementary school; they had to wait until the end to get called. But there actually is something deep here and it is not going to take long but I don't want to lose it.

These definitions are really passing in the night, so it seems. But the big dispute, I think, is actually not that great. Let me tell you what I mean. In the application, in the specification, there are really two different roles that "application community" plays. Think of "application community" as distributed computers. They can be at different locations across the country, they can be in the same room together, but they are running independently. It is a way of using something called parallel computing to unlock the power. You give a piece of a big problem to lots of different workers, and together they solve it.

And there is another concept of "application community" used in which the members of the community all share a same model. And so both these strategies are available and spoken about in the patent. So the patent speaks about "members of the community sharing a common model." So they all run the same model, the model used to test divergence from normal. But they talk about a different, equally important embodiment in which the members of the community don't share a same model. They all contribute to the creation of a model by doing distributed pieces of analysis. And the problem with Symantec's construction is that ignores Option 2. That's really the crux of the dispute.

And you can actually see this as clear as day in the specification. The '115 patent at 6:33 to 36. This is a very important passage in my mind. What the passage says is that "In some embodiments, the members of the application community share models with each other." That's Option 1. There is a model that they share, or more than one model that they share. But they share something in common. The alternative, and/or, they "update each other's models." Interesting. They are not sharing a common model. They are both contributing information. They are contributing information that other members of the community can use to update their own specific model. And this actually, what's interesting is, even though Option 2 -- Option 1, Option 2 are both real and meaningful and important. It is Option 2, the option that Symantec's definition ignores, that gets the most amount of attention in the specification. It is complex, and that's the reason. And sometimes in those situations, you lose the forest for the trees. But they are very, very focused on this Option 2. "Some particular randomly chosen function or functions and its associated data" are divvied out to the different members of the community so they can work on parts of problems, analyze portions of code, determine weaknesses and danger, and then use that information and share it with other members of their community who can create their own models. Option 2. You see that at Column 16, Lines 55 through 58 as well. "Each portion or slice" is divvied out to each member of the workstation.

And so Symantec has this argument where they say, "This construction that Columbia is proposing is fantastically broad and it allows for these absurd," what they characterize as "absurd outcomes." Actually, I think that's a caricature. I don't think that's what's going on here. In our mind the construction we proposed allows for two different options when you read it in light of the claim. Members of the application community either run the same model of application or a portion thereof, or they run an application that allows them to share information that is used to build a model. That's what we believe is the implication of our construction. We don't think it is broad, we don't think it is narrow. We think what it does is, it shows fidelity to the two options that are consistently described in the specification. Thank you very much, Your Honor.

THE COURT: All right. Symantec?

MR. HAMSTRA: Slide 72. Your Honor, going back to basically some of the same content Mr. Sheasby just left off at, in Columbia's responsive brief, they said this here. They believe Columbia's construction requires that the "members of the application" include "those who run the same modeled application or a portion thereof." And that sounds a lot like Symantec's proposed construction of "application community." Assuming that the modeled application refers to the programs in the claims for which there is a model, I don't think we disagree with that at all.

The problem comes from this second statement here, that an "application community" also includes those who "run an application that allows them to share information that is used to build a model."

So the first problem with that statement is that it does not appear supported by Columbia's actual construction. Go to Slide 70. So Columbia's proposed construction is on the right here. It is "members of a community running the same program or a selected portion of a program." There is nothing about that construction that would encompass any application that allows sharing of information to build a model. Slide 72 again. And there are other problems with this statement. For instance, Columbia's statement here just recites a model. What is that model of? Is that a model of the modeled program recited in the claims? We don't know. It doesn't say.

And Columbia referenced the fact that the specification also describes the sharing of models among application community members. But there is a simple reason for that. The application community members share models because they each have a model of that same program, and therefore share pieces of that model.

Flip to Slide 74. So there is a citation in the provisional application here, and again, "application communities are...instances of the same application that...monitor their execution," i.e., the execution of those independent instances, "for flaws and attacks." So they are all looking at that same program that they are running and modeling.

I just want to respond to one point on "anomalous." Columbia has spent a lot of time talking about various methods for detection of malicious software that use some attack data for modeling. But what we haven't seen today a single time is a single quotation describing "a model of normal computer system usage" as a model that includes attack data. That's just something we haven't seen before. I have nothing further, Your Honor.

THE COURT: All right. Rebuttal?

MR. SHEASBY: Your Honor, just one point of rebuttal, which is, I don't believe that last statement was accurate. We, of course, did show you the portion of provisional application as just one example that they omitted from their brief that describes a model of normal data that uses mixed data. With that, I don't think we need to say anything else.

THE COURT: All right.

MR. NELSON: Your Honor, can I address something? Mr. Sheasby has said about five times now that we omitted something from the brief. The provisional application that he is talking about, we submitted that in the Declaration, the first time around. It was 300 pages long. And there was already a number of things in front of Your Honor, right? And what we did is said specifically the parts, the citations that we have in there, that's what we are providing to Your Honor. That was like the first 50 pages. The last 250 pages didn't get cited. They came back then and said, "We want to put the rest of it in." We said, "Fine." We didn't oppose that. So I'm a little bit -- there has been several times where there has been this implication that we were trying to keep something from Your Honor. Absolutely no possible way. That's not what we were trying to do. I will never do that. I will represent that to you as an Officer of the Court. I just want to make sure the record is clear so Your Honor doesn't walk away from here thinking just because Columbia kept saying it, that the Symantec guys are trying to keep information away from Your Honor. That's not the way it is. All right.

THE COURT: All right.

MR. SHEASBY: And Your Honor, I actually agree with that.

THE COURT: Is there anything else anybody wants to say? You all have grand argument?

MR. NELSON: There was only one point that I wanted to make a little bit clear on what Mr. Hamstra said at the end there. The inventor testimony was an example, you saw it at the end. I don't need to put it back up on the screen. Where the inventor said, "Well, we had these models where we included normal data and abnormal data," right? These algorithms. "We had things where we included things in addition to normal data." During my portion of the argument, I showed you that. There were a number of those things that were discussed. But those are different algorithms. That's the point. Different algorithms from the ones that are claimed for various parts in the '084 patent where it says "a model of normal computer usage." That's what they chose to claim. What Mr. Hamstra was saying, the point being made, any of that inventor testimony you saw, the stuff from the provisionals, all these other models there was a distinction made with normal and abnormal data. The one that we have claimed here and the one that was referenced was a model of normal computer system usage, which is one that uses attack-free data. So that was the point that was being made. I just wanted to address that because I think Columbia had a little issue with Mr. Hamstra's point. So thank you.

THE COURT: All right. Okay. I'll take the matter under advisement. It will take me a little while to turn my mind from what I was into and to this, but I will do that shortly and try to get these terms construed so that we can march on. Thank you all very much. I appreciate it.

(Proceedings adjourned at 4:04 p.m.)

CERTIFICATE OF REPORTER

I, Jeffrey B. Kull, Official Reporter, certify that the foregoing is a correct transcript from the record of proceedings in the above-entitled matter.

__________/s/_________________

Jeffrey B. Kull,
Official Federal Reporter

__________/s/_________________

Date